

SOPORTE ORGANIZACIONAL DE MEDICIÓN Y EVALUACIÓN ORIENTADA A OBJETIVOS Y SENSIBLE AL CONTEXTO

Hernán Molina

Grupo de Investigación y Desarrollo en Ingeniería de Software y Web (GIDIS Web)

Calle 9 y 110, (6360) General Pico, La Pampa, Argentina

hmolina@ing.unlpam.edu.ar

Presentada a la Facultad de Informática de la UNLP como parte de los requisitos para la obtención del título de

DOCTOR EN CIENCIAS INFORMÁTICAS

Director: *Dr. Luis OLSINA*

Co-Director: *Dr. Gustavo ROSSI*

La Plata, Junio de 2012

Facultad de Informática
Universidad Nacional de La Plata - Argentina

Resumen

El éxito de un producto o servicio de software depende en gran medida de la satisfacción de un cierto nivel de calidad observado sobre el mismo producto o servicio así como sobre los recursos y procesos involucrados en su obtención y su operación. Conocer de forma cuantitativa y objetiva el nivel de calidad de un producto, servicio, proceso o recurso es fundamental como parámetro para la toma de decisiones hacia un camino de mejora. Sin embargo, lograr esto no es una tarea trivial ya que la calidad es un concepto abstracto y relativo que debe ser considerado desde diversos puntos de vista, involucrando diversos aspectos y en contextos de aplicación particulares.

Para lograr lo anterior, es crucial para una organización contar con programas y procesos de *Aseguramiento de Calidad* para definir las tareas necesarias para detectar y corregir problemas en la calidad así como procesos de *Medición y Evaluación* que provean la información cuantitativa y objetiva sobre los niveles de calidad reales de cada una de las entidades relevantes involucradas. Sin embargo, configurar, ejecutar y mantener un programa de medición y evaluación robusto y consistente no es una tarea simple. Cuando se implementan programas de medición y evaluación en organizaciones de software, ciertos aspectos técnicos clave deben ser resueltos. En primer lugar, se deben establecer claramente, mediante la especificación de un proceso, las actividades a realizar, así como los recursos y los artefactos que serán utilizados y producidos durante su ejecución. Estas actividades deben organizarse de tal manera que puedan ser coordinadas con actividades de soporte relacionadas con actividades ingenieriles. En segundo lugar, se debe establecer de forma explícita un marco conceptual que facilite un entendimiento común de los términos y relaciones utilizados en las actividades mencionadas entre los proyectos de la organización así como el intercambio y reuso consistente de instancias de tales conceptos para que los resultados puedan ser repetibles, comparables y consistentes. En tercer lugar, se deben utilizar métodos, técnicas y herramientas específicos y apropiados para llevar a cabo efectivamente las actividades definidas.

En este sentido existe un número importante de propuestas que definen enfoques, procesos, modelos, métodos y herramientas para llevar a cabo las actividades de medición y evaluación. Sin embargo, la mayoría de estas propuestas no proveen un enfoque integrado que incluya todos estos elementos. Adicionalmente muchas de estas propuestas carece de una base conceptual que defina de forma clara y estructurada los conceptos y relaciones involucradas en tales actividades. Y en los casos donde existe una terminología definida no existe un consenso general (aún entre propuestas de la misma fuente) sobre los términos involucrados que permita integrar diferentes propuestas, complicando aún más los problemas de implementación de programas de medición y evaluación.

En esta tesis se muestran los resultados de una investigación destinada a crear una solución para cubrir dicha falta proponiendo un marco conceptual con base ontológica que define de forma clara y estructurada los elementos de información (los términos y relaciones) utilizados para especificar el diseño e implementación de actividades de medición y evaluación. El marco propuesto fue creado a partir de los términos definidos en la literatura relevante, siguiendo un enfoque orientado a objetivos y enfocado a la organización. Además el marco sigue un enfoque sensible al contexto para proveer un soporte a la toma de decisiones más coherente y consistente entre los proyectos de la organización. El marco resultante conforma una plataforma de soporte para la definición de procesos y métodos para la ejecución de proyectos de medición y evaluación en la organización. La propuesta incluye además una arquitectura de soporte que permite la integración del marco al dominio de aplicación de la organización. Finalmente, estos elementos han sido implementados en una herramienta web que permite visualizar los beneficios de la propuesta.

Agradecimientos

Al Dr. Luis Olsina, por la dirección de esta tesis, y al Dr. Gustavo Rossi, por la co-dirección de la misma;

a todos los colegas del grupo GIDIS Web, por las ideas y el trabajo compartido;

a Pablo Becker, por la lectura y revisión de toda la tesis;

a mi familia y seres queridos que me apoyaron y soportaron todo este tiempo; simplemente, Gracias.

I	INTRODUCCIÓN Y FUNDAMENTOS	15
1.	Introducción	17
1.1.	Motivación	19
1.1.1.	El Rol de la Medición	19
1.1.2.	Modelos para Medición y Evaluación	20
1.1.3.	Enfoque Sensible al Contexto	21
1.1.4.	Problemática	22
1.2.	Metodología de Investigación	24
1.2.1.	Solución Propuesta	25
1.3.	Contribuciones	26
1.4.	Publicaciones relevantes	27
1.5.	Estructura de la Tesis	27
2.	Estado del Arte	29
2.1.	M&E de Calidad en Proyectos de Software y Web	30
2.1.1.	Medición y evaluación como procesos de soporte	30
2.1.2.	Medición y evaluación de calidad	32
2.1.3.	Enfoques, procesos y métodos	43
2.1.3.1.	Enfoques	43
2.1.3.2.	Procesos	46
2.1.3.3.	Métodos	50
2.1.3.4.	Otros enfoques, prácticas y métodos	53
2.1.4.	Elementos de la medición y evaluación	53
2.1.5.	Modelos de información para medición y evaluación	54
2.1.6.	Necesidad de información de contexto en M&E	62
2.2.	Enfoque Sensible al Contexto	67
2.2.1.	Definiciones	68
2.2.2.	Representación	70
2.2.3.	Procesamiento	75
2.2.4.	Aplicaciones	77
2.2.5.	Discusión	79
2.3.	Conclusiones	81

3. Fundamentos	83
3.1. Principios	83
3.1.1. Orientado a Objetivos	83
3.1.2. Basado en metadatos	84
3.1.3. Centrado en la Organización	84
3.1.4. Sensible al contexto	84
3.2. Enfoque de Medición y Evaluación	85
3.2.1. Marco INCAMI y Base Ontológica	85
3.2.2. Proceso de Medición y Evaluación	87
3.2.3. Metodología WebQEM	87
3.2.3.1. Descripción del Modelo LSP	89
3.3. Enfoque Sensible al Contexto Adoptado	90
3.3.1. Requerimientos	91
3.3.2. Caracterización del Contexto	91
3.3.3. Modelado y Representación de Contexto	92
3.3.4. Procesamiento de Información de Contexto	95

II MEDICIÓN Y EVALUACIÓN ORIENTADA A OBJETIVOS Y SENSIBLE AL CONTEXTO **97**

4. El Marco Conceptual C-INCAMI	99
4.1. Introducción al Marco C-INCAMI	99
4.2. Gestión de Proyectos	101
4.3. Especificación de Requerimientos No Funcionales	102
4.3.1. Especificación de Información de Contexto	104
4.4. Definición e Implementación de la Medición	108
4.5. Definición e Implementación de la Evaluación	111
4.6. Integración del Marco C-INCAMI a la Organización	116
5. Medición y Evaluación Sensible al Contexto	121
5.1. Especificación de Contexto y Contextualización	122
5.1.1. Contexto Real	125
5.1.2. Contexto de Aplicación	127
5.2. Uso de Información de contexto	129
5.2.1. Comparación de Contexto	131
5.2.1.1. Similitud Contextual (<i>CSim</i>)	132
5.2.1.2. Aplicabilidad de Entidad Contextual (<i>CApp</i>)	132
5.2.1.3. Similitud de Propiedades de Contexto (<i>CPSim</i>)	134
5.2.1.4. Métricas para Comparación de Contextos	137
6. Prueba de Concepto	141
6.1. Creando un Proyecto de Medición y Evaluación	141
6.2. Especificando los Requerimientos No Funcionales	142
6.2.1. Necesidad de Información	142
6.2.1.1. Descripción del Contexto (Real)	142
6.2.2. Modelo de Concepto (Requerimientos No Funcionales)	148
6.2.2.1. Selección de un modelo de concepto	148
6.2.2.2. Diseño del Modelo de Concepto (Selección de Atributos)	151
6.3. Definir e Implementar la Medición	154

6.3.1.	Diseño de la medición	154
6.3.2.	Implementación de la medición	162
6.4.	Definir e Implementar la Evaluación	162
6.4.1.	Selección de modelos elementales	163
6.4.2.	Selección de un modelo global	166
6.4.3.	Configuración de los indicadores globales y elementales	167
6.4.4.	Selección de los criterios de decisión	169
6.4.5.	Cálculo de indicadores elementales y globales	170
6.5.	Análisis de los Resultados	170
6.6.	Conclusiones	173
7.	Soporte Tecnológico a C-INCAMI	175
7.1.	Alcance	176
7.2.	Arquitectura	176
7.2.1.	C-INCAMI Project Manager	178
7.2.1.1.	Capa de Presentación	179
7.2.1.2.	Capa de Lógica de Negocio	182
7.2.1.3.	Capa de Lógica de Acceso a Datos	195
7.2.1.4.	Capa de Persistencia	200
7.2.2.	Catálogo de M&E	201
7.2.3.	Repositorio del Dominio de Aplicación	202
7.3.	Comentarios finales	202
8.	Conclusiones y Trabajos Futuros	207
8.1.	Revisión de la Propuesta	207
8.1.1.	Elementos de la Propuesta y Problemas Atacados	207
8.1.2.	Características de la Propuesta	211
8.2.	Contribuciones	212
8.3.	Líneas de Investigación Futuras	215
A.	Conceptualización del Modelo de Contexto	217
B.	Esquema RDF del Marco C-INCAMI	221
	Bibliografía	233

Índice de Figuras

1.1. Factores que influyen en el rendimiento organizacional y la calidad de productos y servicios.	18
2.1. Áreas de proceso de soporte básicas.	31
2.2. Integración del proceso de evaluación al ciclo de vida de software según el estándar ISO/IEC 14598-1:1999 (adaptada de dicho estándar).	32
2.3. Modelo de Calidad de McCall.	36
2.4. Modelo de Calidad de Boehm.	37
2.5. Modelo de calidad y calidad en uso del ISO 9126-1.	38
2.6. Modelo de calidad de producto del estándar ISO/IEC 25010 (tomado de [ISO/IEC 25010, 2011]).	38
2.7. Modelo de calidad en uso del estándar ISO/IEC 25010 (tomado de [ISO/IEC 25010, 2011]).	39
2.8. Modelo de calidad de datos del estándar ISO/IEC 25012 (adaptado de [ISO/IEC 25012, 2008]).	39
2.9. Modelo de Experiencia Real de Usuario de Lew et al.	40
2.10. Modelo de calidad de COQUAMO.	41
2.11. Marco conceptual de métricas de calidad del estándar IEEE 1061.	42
2.12. Elementos del enfoque GQM.	45
2.13. Componentes del enfoque BSC.	45
2.14. Modelo de Información de PSM.	55
2.15. Modelo de información de medición presentado en [Kitchenham <i>et al.</i> , 2001].	56
2.16. Modelo conceptual de medición de Briand <i>et al.</i>	57
2.17. Diagrama UML que muestra los conceptos y relaciones de la ontología de medición y evaluación de [Martín & Olsina, 2003].	59
2.18. Diagrama UML que muestra los conceptos y relaciones de la ontología de medición de [García <i>et al.</i> , 2006].	61
2.19. Ontología de medición de [Barcellos <i>et al.</i> , 2010].	62
3.1. Diagrama de clases incluyendo conceptos y relaciones principales del marco INCAMI.	86
3.2. Reuso de las definiciones de conceptos de medición y evaluación en los diferentes proyectos de la organización.	86
3.3. Modelo de Proceso INCAMI.	87
3.4. Grados de conjunción/disjunción de los operadores del modelo LSP.	89
3.5. Caracterización de la información de contexto.	92

3.6. Estructura de alto nivel del modelo de contexto y dependencias con definiciones y datos del dominio de aplicación.	94
4.1. Organización de paquetes o módulos (UML) del marco C-INCAMI.	100
4.2. Diagrama de clases (UML) describiendo los conceptos y relaciones del módulo principal de C-INCAMI.	101
4.3. Diagrama de clases (UML) mostrando los conceptos y relaciones del módulo <i>requirements</i> del marco C-INCAMI.	102
4.4. Diagrama de clases (UML) mostrando los conceptos y relaciones principales del módulo <i>context</i> del marco C-INCAMI.	105
4.5. Diagrama de clases (UML) mostrando las entidades contextuales del módulo <i>requirements</i> del marco C-INCAMI.	106
4.6. Diagrama de clases (UML) mostrando los conceptos y relaciones del módulo <i>measurement</i> del marco C-INCAMI.	109
4.7. Diagrama de clases (UML) mostrando los conceptos y relaciones del módulo <i>evaluation</i> del marco C-INCAMI.	112
4.8. Conceptos del Marco C-INCAMI que se conectan al dominio de aplicación de la organización en repositorios RDF mediante direcciones URI.	117
4.9. Ejemplo de especificaciones C-INCAMI (diagrama de objetos UML) que referencian definiciones en un repositorio RDF (grafo RDF).	118
5.1. Componentes de la situación relativa a una entidad de interés sometida a medición y evaluación a partir de la cual se extrae la información de contexto relevante.	122
5.2. Diagrama de clases (UML) mostrando todos los conceptos y relaciones del módulo <i>context</i> del marco C-INCAMI.	123
5.3. Actividades involucradas en la especificación de los requerimientos no funcionales, incorporando la especificación del <i>contexto real</i>	125
5.4. Actividades involucradas en la especificación del <i>contexto real</i> , durante la especificación de requerimientos no funcionales.	126
5.5. Diagrama de objetos (UML) describiendo un ejemplo de la especificación de un <i>contexto real</i>	127
5.6. Diagrama de objetos (UML) describiendo un ejemplo de la especificación de un <i>contexto de aplicación</i>	129
5.7. Proceso de M&E incorporando los elementos del enfoque <i>sensible al contexto</i>	130
5.8. Diagrama de objetos (UML) ilustrando el uso de las métricas utilizadas en la comparación de contextos reales entre dos proyectos hipotéticos.	138
6.1. Modelos de concepto para Calidad Externa disponibles en el repositorio de la organización.	149
6.2. Modelo de Calidad Externa resultante de la selección de atributos.	155
6.3. Vista parcial del árbol de requerimientos de <i>Calidad Externa</i> mostrando el diseño de indicadores globales y elementales para el modelo LSP (sólo para el concepto calculable <i>Information Quality</i>).	168
6.4. Evaluación del modelo de concepto de <i>Calidad Externa</i>	171
6.5. Diferentes representaciones de los resultados de alto nivel de la evaluación de <i>External Quality</i> de Cuspide.com.	172
6.6. Representación de resultados parciales de la evaluación de <i>External Quality</i> –para los conceptos que componen <i>Suitability</i> (en <i>Information Quality</i>).	173
7.1. Arquitectura de C-INCAMI Project Manager	177

7.2. Páginas HTML estáticas utilizadas como plantillas para la generación de las vistas dinámicas de la herramienta C-INCAMI ^{PM}	180
7.3. Páginas estáticas y enlaces de navegación de la aplicación web C-INCAMI ^{PM}	181
7.4. Componentes que integran el módulo de Workflow del Proceso de M&E.	183
7.5. Diagrama de Casos de Uso de la herramienta C-INCAMI ^{PM} para la Gestión de Proyectos de M&E.	184
7.6. Pantalla de la herramienta C-INCAMI ^{PM} que permite crear, editar y borrar Proyectos de M&E.	185
7.7. Pantalla de la herramienta C-INCAMI ^{PM} que permite gestionar un Proyecto de M&E.	185
7.8. Diagrama de Casos de Uso de la herramienta C-INCAMI ^{PM} para la Especificación de Requerimientos No Funcionales.	186
7.9. Pantalla de la herramienta C-INCAMI ^{PM} que permite editar un modelo de concepto.	187
7.10. Pantalla de la herramienta C-INCAMI ^{PM} que permite seleccionar un atributo para ser agregado a un modelo de concepto.	187
7.11. Diagrama de Casos de Uso de la herramienta C-INCAMI ^{PM} para la Descripción del Contexto Real.	188
7.12. Pantalla de la herramienta C-INCAMI ^{PM} que muestra las propiedades de contexto incluidas en la descripción del <i>contexto real</i>	189
7.13. Pantalla de la herramienta C-INCAMI ^{PM} que permite seleccionar una propiedad de contexto para agregar a la descripción del <i>contexto real</i>	189
7.14. Diagrama de Casos de Uso de la herramienta C-INCAMI ^{PM} para la Gestión de Proyectos de Medición.	190
7.15. Pantalla de la herramienta C-INCAMI ^{PM} que permite asignar una métrica a cada atributo de un modelo de concepto.	191
7.16. Pantalla de la herramienta C-INCAMI ^{PM} que permite registrar una medición para una determinada métrica.	192
7.17. Diagrama de Casos de Uso de la herramienta C-INCAMI ^{PM} para la Gestión de Proyectos de Evaluación	192
7.18. Pantalla de la herramienta C-INCAMI ^{PM} que permite especificar criterios comunes de evaluación.	194
7.19. Pantalla de la herramienta C-INCAMI ^{PM} que permite especificar un indicador elemental.	194
7.20. Pantalla de la herramienta C-INCAMI ^{PM} que muestra los resultados de evaluación para una entidad y concepto dado.	195
7.21. Representación en UML de los objetos instanciados para especificar una necesidad de información.	199
7.22. Diagrama Entidad-Relación del esquema de la base de datos de usuarios de la herramienta C-INCAMI ^{PM}	201
7.23. Extracto del vocabulario del marco C-INCAMI especificado en RDFS.	203
7.24. Algunas descripciones RDF correspondientes a las especificaciones reusables almacenadas en el Catálogo de M&E.	204

Índice de Tablas

2.1. Áreas de Proceso de CMMI que dependen directamente del área de proceso Medición y Análisis.	31
5.1. Especificación de la métrica “Similitud Contextual” (<i>CSim</i>) usando la definiciones provistas por el marco C-INCAMI.	133
5.2. Especificación de la métrica “Aplicabilidad de Entidad Contextual” (<i>CApp</i>) usando las definiciones provistas por el marco C-INCAMI.	134
5.3. Especificación de la métrica “Similitud de Propiedad de Contexto” (<i>CPSim</i>) usando la definiciones provistas por el marco C-INCAMI.	135
5.4. Método de medición asociado al <i>Criterio de Similitud de Exacto</i> utilizando el operador <i>AND</i> (<i>ESC_{AND}</i>).	136
6.1. Datos del Proyecto de M&E.	141
6.2. La especificación de la <i>Necesidad de Información</i> para la evaluación de Calidad externa de la aplicación web Cuspide.com	142
6.4. Criterio de similitud para la propiedad de contexto “Automated Support”.	146
6.5. Métrica que cuantifica la propiedad de contexto “ <i>Application domain</i> ”.	146
6.6. Métrica que cuantifica la propiedad de contexto “ <i>Entity category</i> ”.	147
6.7. Métrica que cuantifica la propiedad de contexto “ <i>Target market scope</i> ”.	147
6.8. Métrica que cuantifica la propiedad de contexto “ <i>Supported natural language</i> ”.	148
6.9. Contextos de aplicación de los Modelos de Concepto de la Figura 6.1.	150
6.10. Niveles de aplicabilidad contextual al proyecto de M&E para los modelos de concepto de <i>Calidad Externa</i> encontrados en el catálogo de M&E.	151
6.11. Atributo “Shopping cart control ease to be recognized”.	152
6.12. Atributo “Readability of text”.	153
6.13. Atributo “Identification of content provider”.	153
6.14. Atributo “Foreign Language Support”.	154
6.15. Métrica “Fog Index” para cuantificar el atributo “Readability of Text”.	157
6.16. Métrica “Lexile Text” para cuantificar el atributo “Readability of Text”.	158
6.17. Métrica “ <i>Degree of completeness to the line item information</i> ” (v1.0) para cuantificar el atributo “ <i>Line item information completeness</i> ”.	159
6.18. Métrica “ <i>Degree of completeness to the line item information</i> ” (v2.0) para cuantificar el atributo “ <i>Line item information completeness</i> ”.	160

6.19. Especificación de la Métrica “ <i>Available shipping and handling information</i> ” para cuantificar el atributo “ <i>Shipping and handling information completeness</i> ”.	161
6.20. Métrica “ <i>Supported Natural Languages</i> ” para cuantificar el atributo “ <i>Foreign Language Support</i> ”.	161
6.21. Métrica “ <i>Amount of Supported Natural Languages</i> ” para cuantificar el atributo “ <i>Foreign Language Support</i> ”.	161
6.22. Mediciones realizadas sobre la entidad <i>Cuspide.com</i> para las métricas vistas en la Subsección 6.3.1.	162
6.23. Contexto de aplicación del Modelo Elemental EM_{SNL1} que evalúa al atributo <i>Foreign Language Support</i>	164
6.24. Contexto de aplicación del Modelo Elemental EM_{SNL2} que evalúa al atributo <i>Foreign Language Support</i>	164
6.25. Contexto de aplicación del Modelo Elemental EM_{SNL3} que evalúa al atributo <i>Foreign Language Support</i>	165
6.26. Criterio de decisión <i>External certification criteria</i>	169
6.27. Criterio de decisión <i>Internal evaluation criteria</i>	170
7.1. Clases incluidas en los subpaquetes de <i>gidisweb.c_incami.repository</i>	196

Parte I

INTRODUCCIÓN Y FUNDAMENTOS

Introducción



En la actualidad, el desarrollo y mantenimiento de sistemas de software son actividades corrientes en el funcionamiento de muchas empresas y organizaciones. Ya sea como actividad principal o como soporte a las mismas, las organizaciones crean y/o mantienen sistemas de software muchas veces críticos para su funcionamiento. Por tal razón, actividades tales como la *planificación, análisis, diseño, implementación, prueba y mantenimiento* de productos de software deben ser gestionadas eficientemente siguiendo un conjunto de procesos en el marco de uno o más proyectos, haciendo uso de una cantidad de recursos limitados, bajo ciertas condiciones y restricciones. En este escenario, el éxito de estas organizaciones depende en gran medida de la calidad alcanzada por los productos obtenidos manteniendo al mismo tiempo un cierto nivel de rendimiento en su producción. Tal como se refleja en la Figura 1.1 (adaptada de [Pressman, 2001]), estos factores de éxito están directamente influenciados por la complejidad de los requerimientos de esos productos, las características de los métodos y herramientas utilizadas para desarrollarlos, las personas involucradas en el proceso (tanto clientes como integrantes del grupo de desarrollo) y un conjunto de factores y condiciones de ambiente que relacionan entre sí todos estos elementos.

Lograr un balance entre estos factores es un desafío no menor que requiere llevar a cabo un control preciso de los parámetros involucrados. Este control involucra un seguimiento de un conjunto de atributos e indicadores que reflejen:

- la satisfacción de los requerimientos de los productos,
- el cumplimiento de las restricciones de negocio, de recursos y de cronogramas,
- el uso efectivo y eficiente de los métodos,
- las características relevantes y el uso efectivo y eficiente de las herramientas y plataformas de soporte,
- el compromiso y el desempeño de las personas involucradas en el proyecto,
- las características relevantes de los clientes.

Un control efectivo de estos parámetros sólo es posible si se cuenta con información cuantitativa basada en hechos que describa los atributos clave de las entidades involucradas tales como productos, recursos, proyectos y procesos. Contar con esta información permitirá evaluar el estado de dichas entidades en base a un conjunto de indicadores que describan el desempeño esperado. De esta forma se podrán

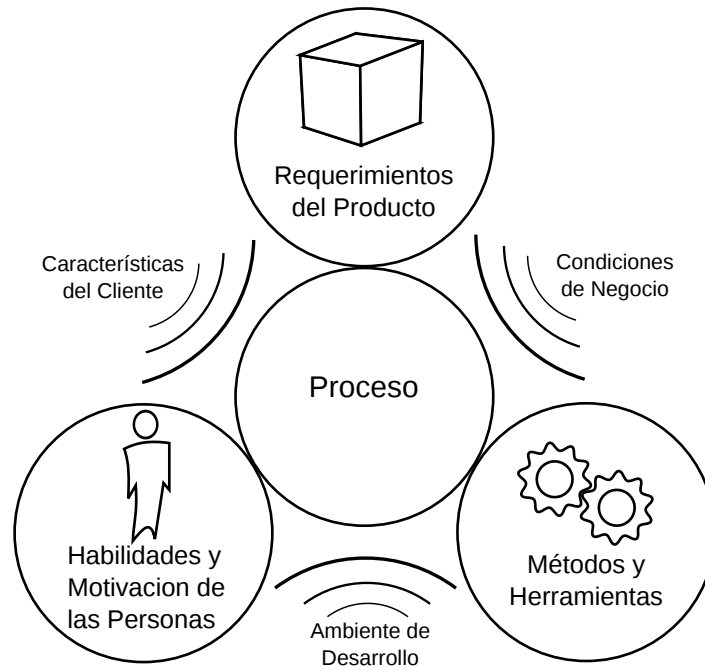


Figura 1.1: Factores que influyen en el rendimiento organizacional y la calidad de productos y servicios.

mejorar las estimaciones de los resultados, gestionar y verificar objetivamente el cumplimiento de los objetivos de la organización y planificar las acciones correctivas necesarias de forma más precisa. Decisiones tomadas en ausencia de estas condiciones incrementan la probabilidad de error en los resultados esperados pudiendo llevar a la organización a alejarse aun más de su verdadero objetivo.

Para alcanzar tal nivel de información se debe contar con modelos que permitan representar de forma clara y explícita los factores involucrados en la determinación del estado actual de proyectos, procesos, recursos y productos de forma que los datos relevantes puedan ser registrados, accedidos y utilizados para llevar a cabo el seguimiento que permitirá a la organización alcanzar sus objetivos de forma eficiente. Estos modelos deben permitir representar los diversos aspectos de calidad de las entidades involucradas, los valores cuantitativos que reflejen el estado real de las mismas, y la información necesaria para interpretar dichos datos para ser utilizados en la toma de decisiones gerencial.

En esta tesis se propone un marco conceptual de medición y evaluación (en adelante *M&E*) que define de forma clara y estructurada los términos y relaciones utilizados para especificar el diseño e implementación de dichas actividades. Todos los elementos del marco propuesto se encuentran definidos semánticamente ya que posee una base ontológica. El marco sigue un enfoque orientado a objetivos y está enfocado a la organización al proveer los términos que le permiten incorporar la medición y evaluación como un proyecto de la misma. Un aspecto de la contribución a destacar es la aplicación de un enfoque sensible al contexto, independiente del dominio, para proveer una solución más coherente y consistente al diseño, implementación, interpretación y, en consecuencia, a la toma de decisiones en proyectos de medición y evaluación en una organización. Además se define una arquitectura de soporte al marco propuesto que permite su integración al dominio de aplicación de la organización con un esfuerzo mínimo. El marco conceptual y la arquitectura propuestas constituyen una plataforma de soporte a la definición de procesos, métodos y herramientas para la ejecución de proyectos de medición y evaluación en una organización. Así, finalmente se presentan detalles de una herramienta web construida para dar soporte al marco propuesto mostrando una posible instanciación de la arquitectura diseñada para visualizar los beneficios de la propuesta.

A continuación se presentan los aspectos que motivaron la realización de esta investigación describiendo brevemente el rol de la medición y evaluación en el desarrollo de productos de software en una organización, el uso de modelos como mecanismo para guiar dichas actividades así como la descripción

puntual de la problemática encontrada en los mismos, que será atacada en esta tesis. Luego se describe la solución propuesta enumerando las contribuciones realizadas y las publicaciones relevantes que lo anteceden.

1.1. Motivación

1.1.1. El Rol de la Medición

La medición tiene como propósito primario conocer de forma objetiva el estado de una entidad (es decir, sus atributos) de interés para un propósito y un punto de vista en particular. Cabe aclarar que cuando se habla de medición, en cualquier área de la ciencia, se asume la existencia de la evaluación o análisis de los valores medidos, ya que no se concibe la medición de atributos de una entidad por el simple hecho de recolectar datos, sino más bien para ser interpretados y analizados en la toma de decisiones destinadas a mantener un cierto nivel de satisfacción en los diversos aspectos de dichas entidades. Así, en la ingeniería de software, los procesos de medición y evaluación, definen las actividades necesarias para planificar, diseñar, recolectar, almacenar y analizar los resultados de mediciones cuantitativas de entidades tales como procesos, proyectos, recursos y productos de software. Como se mencionó en la introducción, un aspecto particularmente importante de una entidad en el dominio de la Ingeniería de Software es su calidad, factor determinante en el éxito de todo producto o servicio de software. En este sentido, cabe destacar la dificultad inherente de definir este concepto para una entidad dada, ya que se trata de un concepto relativo que depende del punto de vista desde el que se lo considera, entre otros, y que está compuesto de diversos aspectos o atributos de calidad, en su mayoría abstractos y/o no visibles desde una perspectiva de un agente externo.

En líneas generales, las mediciones permiten atender a cuatro necesidades en una organización [Park *et al.*, 1996]:

- para caracterizar y comprender el estado de procesos, productos, recursos y ambientes, y para establecer valores de referencia para realizar comparaciones con mediciones futuras;
- para evaluar el estado actual de procesos y productos respecto de los parámetros previstos o planificados y controlar que vuelvan a los valores esperados si se salen de los mismos, así como para evaluar la satisfacción de los objetivos de calidad y el impacto de las mejoras de procesos y tecnologías en productos y procesos;
- para predecir las propiedades clave de las entidades relevantes y planificar en función de ellas, para lo cual se utilizan mediciones de procesos y productos para comprender las relaciones entre ellos y construir modelos de sus dependencias. De esta forma, los valores observados de ciertos atributos pueden ser usados para predecir otros relacionados. Las predicciones permiten establecer objetivos alcanzables de costos, cronogramas y atributos de calidad, y así, aplicar los recursos apropiados a tal fin. Las mediciones predictivas también pueden ser usadas como base para extrapolar tendencias y actualizar estimaciones de costo, tiempo y calidad basándose en evidencia real. Las proyecciones y estimaciones basadas en datos históricos ayudan a analizar riesgos y controlar costos;
- para mejorar la calidad de los productos o el desempeño de los procesos identificando problemas, ineficiencias y otras oportunidades de mejora, así como para planificar esfuerzos de mejora. Las mediciones de desempeño actual proveen una guía de base para realizar comparaciones y determinar si las acciones de mejora están funcionando e identificar efectos colaterales.

Para que los esfuerzos de llevar a cabo las actividades mencionadas arriba sean rentables, la medición y evaluación de procesos, proyectos, recursos y productos debe ser diseñada y dirigida para soportar

los objetivos de negocio de la organización. De esta forma, si tales actividades son correctamente implementadas, los procesos de medición y evaluación pueden soportar el resto de los procesos y prácticas específicas del dominio de la organización, proveyendo una plataforma objetiva para la toma de decisiones que permitan guiar sus proyectos y a la organización misma hacia la satisfacción de los objetivos de alto nivel.

1.1.2. Modelos para Medición y Evaluación

Para que los esfuerzos de medición y evaluación puedan servir a sus propósitos los resultados que surgen de planificar, diseñar y ejecutar estas actividades deben poder ser especificados, almacenados, comunicados y procesados de forma consistente. Datos tales como la necesidad de información –por la cual se lleva a cabo la medición y evaluación–, las entidades que serán medidas y sus atributos, los aspectos o dimensiones de la calidad que serán evaluadas, los mecanismos que definen cómo se obtendrán los valores y las mediciones e interpretaciones realizadas deben ser especificadas, registradas y comunicadas de forma consistente, coherente, clara, y estructurada. Para ello se requieren modelos que definan aquellos conceptos, y las relaciones entre ellos, involucrados en la especificación de los elementos de información mencionados. Estos modelos pueden ser utilizados luego para estructurar datos de medición y evaluación en repositorios tales como catálogos o memorias de experiencias que permitirán a la organización almacenar los datos particulares de cada proyecto y así comunicar efectivamente el estado de su avance y de los productos y recursos involucrados. De esta forma se dispondrá de los datos necesarios para efectuar evaluaciones o comparaciones entre mediciones actuales y pasadas, útiles para efectuar estimaciones, evaluaciones de progreso, entre otros.

Un modelo de medición y evaluación permite enlazar las necesidades de información con las entidades y atributos de interés. El modelo debe describir cómo los atributos relevantes son cuantificados y cómo convertir dichos valores a indicadores para proveer una base para la toma de decisiones [ISO/IEC 15939, 2001]. En este sentido, el modelo debe permitir la especificación de los metadatos que describen las propiedades de los elementos de información reusables, necesarios tanto para diseñar los requerimientos, la medición y la evaluación, así como para su implementación. Además el modelo debe permitir registrar los datos que surgen como resultado de la correspondiente implementación, es decir, los valores obtenidos de la medición de los atributos y de la interpretación de los mismos. En líneas generales, el modelo debería permitir especificar los siguientes elementos de información:

- Especificaciones de la necesidad de información u objetivos de la medición y evaluación;
- Descripciones de las entidades a medir y evaluar;
- Especificaciones de los atributos de dichas entidades;
- Descripciones de contextos en los cuales se lleva a cabo la medición y evaluación;
- Modelos de calidad que representen una visión particular de los requerimientos establecidos por la necesidad de información (por ejemplo, *Confiabilidad, Eficiencia, Mantenibilidad, Efectividad, Satisfacción*, etc.);
- Especificaciones de los mecanismos o métodos a utilizar para recolectar los valores de los atributos, así como descripciones de la forma de representación de sus valores (escalas, unidades);
- Los valores medidos así como descripciones de las propiedades asociadas a los mismos (por ejemplo responsable de la recolección, tiempo y hora de recolección, etc);
- Especificaciones de los mecanismos o métodos a utilizar para interpretar los valores medidos de los atributos así como su agrupación en conceptos de alto nivel que satisfacen la necesidad de información enunciada;

- Los valores correspondientes a la interpretación de los valores medidos.

Esta información debe ser consistente y coherente para servir a todos los niveles de decisión de la organización, desde los niveles inferiores en un proyecto hasta el más alto nivel gerencial de la empresa. Por lo anterior, es evidente que un modelo de medición y evaluación con las características expuestas anteriormente es un requerimiento crítico en la implantación de tales actividades en una organización; de allí la motivación de esta tesis en proveer una propuesta que atienda a tal necesidad.

1.1.3. Enfoque Sensible al Contexto

De los elementos que debieran componer un modelo de medición y evaluación, enumerados en la subsección anterior, se debe resaltar el de contar con una representación clara y estructurada de las descripciones del contexto relevante en el cual se encuentran las entidades a medir, ya que este puede afectar la forma en que se lleva a cabo la medición así como las interpretaciones que surjan de los resultados obtenidos [Park *et al.*, 1996; ISO/IEC 15939, 2001; Basili *et al.*, 1994; Basili & Rombach, 1988; Florac *et al.*, 1997] .

El tratamiento de información de contexto es el objetivo principal del enfoque *sensible al contexto* (*context-aware*) que provee un conjunto de guías que permiten implementar, para cualquier dominio de aplicación, los mecanismos necesarios para representar y procesar descripciones de los contextos relevantes a fin de ser utilizadas para mejorar la coherencia en las actividades que se desarrollan en el mismo.

En este punto, cabe realizar una aclaración respecto de la interpretación dada al término "*sensible al contexto*" utilizado en esta tesis. El calificativo, originalmente en inglés, "*context-aware*" (pudiéndose traducir como "*consciente del contexto*") fue acuñado para describir a aquellas aplicaciones que conocen (son conscientes de) el contexto en el cual se encuentran inmersas y, por lo tanto, usan tal información para actuar o funcionar acorde a tal conocimiento. En este trabajo se consideró más apropiado el término "sensible al contexto" ya que no solo implica el hecho de conocer el contexto sino también de actuar en función del mismo.

Aunque el enfoque fue utilizado por primera vez en el campo de la computación ubicua, donde la influencia del contexto es evidentemente determinante del comportamiento de una aplicación móvil, y dada la interpretación anterior del término, se sostiene que el enfoque puede aplicarse a cualquier dominio donde la información de contexto relevante influya en el comportamiento de la aplicación. Por esta razón no se habla de "*sensible al contexto*" como un área o dominio de aplicación sino más bien como un enfoque que puede ser aplicado a cualquier área, ya sea en computación ubicua, inteligencia artificial, gestión del conocimiento, medición y evaluación, etc. No obstante, algunas propuestas tratan de forma indiferente al enfoque sensible al contexto y al dominio de la computación ubicua [Dey, 2000; de Freitas Bulcão Neto & da Graça Campos Pimentel, 2005; Baldauf *et al.*, 2007; Chen & Kotz, 2000].

Para aplicar el enfoque *sensible al contexto* a un dominio de aplicación se deben proveer ciertos elementos. Por un lado, se debe contar con un modelo que permita representar (i) descripciones de contexto para el dominio de aplicación y (ii) la contextualización de los elementos del dominio. Este modelo debe ser simple pero lo suficientemente expresivo, tener una base conceptual claramente definida y utilizar una representación acorde al propósito y a los requerimientos de procesamiento. Dado que el uso del término contexto es relativamente nuevo en el modelado conceptual de sistemas, la construcción de un modelo para representar información de contexto requiere un análisis previo de su significado. En este sentido muchos autores concuerdan con la definición provista por [Dey, 2001] de que "contexto es cualquier información que puede ser utilizada para caracterizar la situación de una entidad. Una entidad es una persona, lugar u objeto que es considerado relevante en la interacción entre un usuario y una aplicación, incluyendo el usuario y la aplicación mismos". Aunque esta definición no sea completa (como se discutirá en el Capítulo 4) puede ser utilizada como punto de partida para entender qué es contexto y, por lo tanto, cómo modelarlo en la implementación de un enfoque sensible al contexto.

Adicionalmente, la aplicación del enfoque sensible al contexto requiere un análisis previo de cuáles son las dimensiones o factores de contexto relevante, que serán utilizadas en las descripciones de contexto, ya que estas varían dependiendo del dominio de aplicación. Mientras en el dominio de la computación ubicua las dimensiones Rol, Ubicación, Dispositivo, etc., son apropiadas para describir el contexto relevante de un usuario móvil, en el dominio de la Ingeniería de Software las dimensiones Proceso, Proyecto, Producto, Recurso, Restricción, etc. son más apropiadas para describir el contexto relevante de alguna entidad involucrada en el ciclo de vida de desarrollo de software. Por lo tanto, las propiedades o atributos utilizadas para describir el contexto en un dominio de aplicación dado provendrán de las especificaciones y conceptualizaciones de dicho dominio.

Por último, se deben definir mecanismos para la captura/adquisición de la información relevante del contexto así como para el procesamiento de la misma, necesario para ajustar el funcionamiento de la aplicación al estado del contexto. Este procesamiento requerirá de un mecanismo de comparación entre descripciones de contextos, ya sea entre contextos reales o entre un contexto real y un contexto esperado.

Todos estos elementos deberán ser conjugados en una arquitectura tecnológica que permita materializar los modelos y mecanismos diseñados. Esta arquitectura debe ser integrada al sistema o aplicación en el cual se aplica el enfoque.

En la literatura existen diversas definiciones y aproximaciones del enfoque sensible al contexto, aplicadas a diversas áreas de la ciencia. Y aunque todas ellas están de acuerdo en una definición general del término, proveen diferentes soluciones para el modelado y el procesamiento de la información de contexto. En muchos casos proponen modelos basados en caracterizaciones generales de factores de contexto tales como Ubicación, Actividad, Actor, Dispositivo y Tiempo [Zimmermann *et al.*, 2007; de Freitas Bulcão Neto & da Graça Campos Pimentel, 2005; Baldauf *et al.*, 2007]. En otros casos, los modelos están basados en modelos o metamodelos que permiten adaptar las descripciones de contexto al dominio particular de aplicación [Fuchs *et al.*, 2005; Lonsdale & Beale, 2004; Kashyap & Sheth, 1997; Huang & Tao, 2004; Strang *et al.*, 2003; Grossniklaus & Norrie, 2007] o una mezcla de ambos enfoques, donde se provee un conjunto de dimensiones predefinidas con la posibilidad de enlazar a estos conceptos de dominio específico [de Freitas Bulcão Neto & da Graça Campos Pimentel, 2005; Gu *et al.*, 2004; Kaltz *et al.*, 2005; Klemke, 2002].

1.1.4. Problemática

En la literatura relacionada existen diversas propuestas de modelos para medición y evaluación, sin embargo todas ellas carecen de algunas de las características deseables, introducidas en la sección anterior y descriptas a continuación:

- No se definen de forma clara y estructurada los conceptos y relaciones: en el mejor de los casos las propuestas incluyen la definición de un conjunto de términos y relaciones pero sin detallar cómo estos deben ser especificados, es decir, cuáles son los atributos o propiedades que los describen de forma concreta. En muchos casos el conjunto de términos incluidos en el modelo son insuficientes para especificar todos los aspectos requeridos de la medición y evaluación –por ejemplo, no incluyen aquellos términos necesarios para interpretar (computar los valores correspondientes) los factores o atributos de calidad de alto nivel a partir de un conjunto de métricas; en otros casos ni siquiera se provee la definición de los términos necesarios para registrar de forma concisa la necesidad de información por la cual se lleva a cabo la medición.
- Las definiciones son incompletas e inconsistentes entre propuestas de diferentes autores; a veces incluso entre propuestas de la misma fuente; el ejemplo más claro y más conflictivo es el uso de los términos “medida” y “métrica” para describir el mecanismo utilizado para obtener el valor de un atributo. Este problema dificulta la implementación de una solución integral de un programa de

medición y evaluación, ya que existen pocas propuestas que cubran todos los aspectos necesarios, por lo que se debe recurrir a integrar soluciones de diferentes fuentes y autores.

- No disponen de una base conceptual sólida, por ejemplo una ontología, que defina de forma clara los términos, propiedades y relaciones en el dominio de la medición y evaluación de calidad de software. Esta conceptualización es clave para contar con un cuerpo consensuado de los términos utilizados que permita la comunicación e interpretación sin ambigüedades de las especificaciones de medición y evaluación.
- No se contempla la especificación estructurada de información de contexto; no existe, al momento de la escritura de esta tesis, una propuesta de modelo de medición y evaluación que incorpore de forma estructurada, así como con los términos propios del dominio (métricas, medidas, atributos, etc), la especificación de la información de contexto necesaria para establecer un marco coherente en el diseño e interpretación de los resultados de evaluaciones. No obstante la mayoría de los trabajos relacionados reconocen la necesidad de conocer y especificar el contexto de las entidades evaluadas para obtener resultados más coherentes, pero sin proponer un modelo que permita registrar y utilizar esta información en el diseño e interpretación de proyectos de medición y evaluación.
- No se provee un enfoque claro de integración a la organización; uno de los aspectos críticos en el éxito de todo programa de medición es la factibilidad por parte de las organizaciones de concretar su implementación e incorporación a su infraestructura de procesos y proyectos. Esta dificultad puede ser atenuada desde el diseño del modelo conceptual, al incorporar ciertos elementos que faciliten la implementación de mecanismos de integración a los sistemas de la organización. En la literatura relacionada esta dificultad es reconocida y tratada proveyendo guías y recomendaciones sobre cómo incorporar las actividades de medición y evaluación en la organización aunque dejando de lado el aspecto técnico de la adopción de un modelo de medición y evaluación como herramienta crucial en el logro de tal propósito.

Respecto de la aplicación de un enfoque sensible al contexto ya sea independiente del dominio o para el dominio de la Ingeniería de Software, existen también diferentes propuestas, aunque todas ellas carecen de alguna/s de las características deseables para esta propuesta, a saber:

- la información de contexto debe poder ser validada semánticamente contra un modelo de dominio;
- se requiere de un mecanismo de cuantificación robusto para las propiedades de contexto que permita una validación y comparación consistente de las especificaciones de contexto;
- el modelo debe ser independiente del dominio y debe poder ser adaptado para cubrir cualquier dominio de aplicación;
- el modelo debe ser lo suficientemente simple para ser especificado y procesado manteniendo un balance entre costo y desempeño en el sistema;
- el modelo de contexto debe poder ser integrado a sistemas existentes con el menor esfuerzo.s

A la luz de los problemas identificados en las propuestas existentes, en este trabajo se ha propuesto una solución que intenta mitigar las carencias enunciadas arriba y que se describe de forma general a continuación.

1.2. Metodología de Investigación

El objeto de estudio de la investigación realizada en el marco de esta tesis es la representación explícita y estructurada de los metadatos y datos utilizados en el diseño e implementación coherente y consistente de las actividades de M&E de requerimientos no funcionales en proyectos de software y web. Entre estos metadatos y datos se destacan aquellos que permiten especificar de forma efectiva:

- los objetivos de M&E y los requerimientos no funcionales;
- la descripción del contexto relevante en el que se llevan a cabo las actividades mencionadas;
- el diseño e implementación de la medición de los requerimientos no funcionales de bajo nivel;
- el diseño e implementación de la evaluación de los requerimientos no funcionales de bajo y alto nivel.

La hipótesis que subyace a esta investigación plantea que el marco conceptual que se propone en esta tesis permite especificar de forma explícita, estructurada, consistente y coherente los metadatos y datos necesarios para llevar a cabo las actividades de M&E de requerimientos no funcionales en proyectos de software y web (enumerados anteriormente). En este sentido, se asume que contar con los metadatos y los datos claramente especificados, así como con la información de contexto especificada de la misma forma, constituyen elementos clave para lograr resultados de M&E coherentes y consistentes. Estos argumentos son apoyados por un número de resultados previos de otros autores, publicados en la literatura relacionada, basados en experiencias y considerados de relevancia suficiente para ser utilizados como supuestos válidos. Concretamente, dichos resultados exponen lo siguiente:

- Para que los resultados que surjan de la definición, recolección y uso de datos en procesos de medición y evaluación sean efectivamente utilizables y confiables es necesario contar con un modelo de información que permita definir claramente los valores resultantes de la medición y evaluación así como los metadatos que los describen [Card, 2000; Kitchenham *et al.*, 2001; Lawler & Kitchenham, 2003; ISO/IEC 15939, 2001].
- La información del contexto relevante a las actividades de medición y evaluación de calidad en proyectos de software en una organización debe ser especificada de forma explícita y asociada a cada elemento de información cuya aplicación y/o interpretación sea sensible al contexto donde se aplique/interprete. Además dicha información debe ser almacenada junto a la información del proyecto (o elemento de información) que caracteriza para su uso futuro en evaluaciones y/o comparaciones. La disponibilidad de esta información de contexto aporta coherencia a la implementación e interpretación de los resultados correspondientes a las actividades de medición y evaluación. [Briand *et al.*, 2002; Bailey *et al.*, 2003; Kitchenham *et al.*, 2007; ISO/IEC 15939, 2001; Park *et al.*, 1996; Bailey & Basili, 1981; Basili *et al.*, 1994; Basili & Rombach, 1988; Gresse *et al.*, 1995; Card, 2000; Kitchenham *et al.*, 2001; Bevan, 1999; Florac *et al.*, 1997]

Las descripciones de contexto forman parte de la información utilizada para definir e implementar actividades de medición y evaluación, por lo que el primer supuesto se aplica también a la información de contexto.

En el Capítulo 2 se revisan los argumentos de las publicaciones citadas anteriormente, que dan fundamento al marco teórico en el que se apoya la hipótesis planteada en esta tesis.

1.2.1. Solución Propuesta

En esta tesis se describe una propuesta para el soporte conceptual y tecnológico a los procesos de medición y evaluación, siguiendo un enfoque orientado a objetivos y sensible al contexto, en organizaciones que lleven a cabo proyectos de software.

El desarrollo de la investigación presentada en esta tesis se llevó a cabo siguiendo los siguientes pasos:

Se comenzó con la búsqueda y análisis de la literatura relacionada, tanto a la representación de información de M&E y relacionados, así como a enfoques sensibles al contexto, su representación y su aplicación a diferentes áreas de dominio, prestando especial atención al área de ingeniería de software.

Luego se trabajó en la propuesta objeto de esta tesis para cubrir las necesidades planteadas. La propuesta se construye sobre resultados de investigaciones previas que incluyen:

- El marco conceptual de medición y evaluación INCAMI (por *Information Need, Concept model, Attribute, Metric and Indicator*) [Olsina *et al.*, 2008b] que define los conceptos y relaciones necesarios para especificar tanto los datos como los metadatos que surjan del diseño e implementación de actividades de medición y evaluación de los artefactos involucrados en el desarrollo de aplicaciones de software, como procesos, proyectos, recursos, productos, etc. El marco INCAMI soporta la medición y evaluación cuantitativa de entidades siguiendo un enfoque orientado a objetivos, por el cual, todas las actividades son guiadas por una especificación inicial de la necesidad de información. Adicionalmente el marco soporta la especificación de proyectos de medición y evaluación como punto de partida para tareas de gestión de tales actividades.
- La ontología de Métricas e Indicadores [Olsina *et al.*, 2008c] que define formalmente los términos y relaciones utilizados en el marco INCAMI.
- Un catálogo de Métricas e Indicadores [Molina *et al.*, 2004] construido con tecnología de web semántica que permite almacenar, consultar y recuperar definiciones de términos reusables (metadatos) e instancias de los mismos (datos).

La solución propuesta en esta tesis consta de tres elementos principales:

- Un marco conceptual de medición y evaluación orientado a objetivos y sensible al contexto, denominado C-INCAMI (por *Contextual INCAMI*), que permite especificar los metadatos y datos de los conceptos involucrados en el diseño e implementación de dichas actividades. Los conceptos del marco incluyen las especificaciones necesarias para integrar los datos y metadatos del dominio de aplicación a las especificaciones del diseño e implementación de proyectos de medición y evaluación. Todos los términos y relaciones del marco C-INCAMI se encuentran definidos en una ontología.
- Un conjunto de métricas a ser utilizadas en la comparación de especificaciones de contexto que conforman el núcleo de un mecanismo de recomendación sensible al contexto
- Una arquitectura de soporte al marco propuesto que facilita su integración a los sistemas de la organización que incluye:
 - Un catálogo que contiene metadatos y datos reusables de los términos utilizados en el diseño e implementación de la medición y la evaluación de entidades.
 - Una herramienta web para gestionar proyectos de medición y evaluación manteniendo una memoria de proyectos pasados. Esta herramienta incorpora las métricas de comparación de contexto para implementar el sistema de recomendación sensible al contexto. El sistema provee dos tipos de recomendaciones: (i) durante el diseño de medición y evaluación, el

sistema recomienda el uso de los conceptos más apropiados para cada caso en función a su aplicabilidad al contexto del proyecto –por ejemplo, durante la selección de métricas para cuantificar los atributos de un modelo de calidad- y (ii) el sistema recomienda acerca de la validez de comparar los resultados de dos proyectos diferentes en base a la similitud de sus contextos.

A continuación se desarrolló un ejemplo de aplicación de los elementos mencionados mediante una prueba de conceptos para mostrar la factibilidad de la propuesta. La prueba de conceptos desarrollada ilustra el uso de las definiciones (conceptos, propiedades y relaciones) del marco conceptual propuesto, así como de los mecanismos de procesamiento asociados, para llevar adelante la especificación de los metadatos y datos involucrados en todas las actividades del proceso de medición y evaluación.

Contar con el marco y la arquitectura asociada, tal como se proponen en este trabajo, permitiría a una organización implementar programas de medición y evaluación reduciendo el esfuerzo de puesta en marcha asociado a aspectos técnicos y enfocándose en las actividades centrales de definir e implementar la medición y evaluación de las entidades relevantes para satisfacer las necesidades de información que posibiliten el alcance de los objetivos de la organización.

1.3. Contribuciones

A continuación se enumeran las principales contribuciones de esta tesis:

- Se incorporan al marco INCAMI los metadatos y mecanismos que permiten integrar la información del dominio de aplicación de una organización a las especificaciones de M&E realizadas con el marco C-INCAMI. Este mecanismo está basado en tecnologías de la web semántica.
- Se define un modelo de contexto independiente del dominio, utilizando como base conceptual los términos de la teoría de medición definidos en el marco INCAMI; de esta forma el modelo de contexto hereda el mecanismo de cuantificación para describir propiedades de contexto de forma más robusta, semántica y sintácticamente, así como el mecanismo de integración que permite adaptar el modelo de contexto al dominio de aplicación de la organización. Además se diseña un conjunto de métricas que permiten realizar comparaciones de contextos y que conforman el núcleo de un sistema de recomendación sensible al contexto.
- Se aplica al marco INCAMI el enfoque sensible al contexto utilizando el modelo de contexto definido para permitir la especificación del contexto relevante en el cual se realiza la medición y evaluación así como del contexto relevante en el cual son aplicables las definiciones de términos reusables de medición y evaluación. Esta información de contexto será útil en la recomendación de especificaciones reusables en el diseño de medición y evaluación así como en la interpretación y comparación de resultados de diferentes proyectos.
- La extensión de la ontología de métricas e indicadores para incluir los términos y relaciones asociados a la especificación de contexto.
- El diseño e implementación de una arquitectura que integra la propuesta de soporte de medición y evaluación basada en el marco conceptual resultante. La arquitectura incluye (i) una herramienta web que permite especificar los datos y metadatos de medición y evaluación, como un proyecto de la organización, con capacidades de recomendación sensible al contexto, (ii) un catálogo de definiciones de términos de medición y evaluación reusables y (iii) las conexiones y dependencias con los repositorios existentes de la organización que almacenan los datos correspondientes al dominio de aplicación de la misma.

1.4. Publicaciones relevantes

- **Molina, H.**; Olsina, L. & Rossi, G. (2010), Context-Based Recommendation Approach for Measurement and Evaluation Projects. En *Journal of Software Engineering and Applications*, (Scientific Research Publishing), tomo 3(12); págs. 1089–1106, ISSN 1945-3116.
- **Molina, H.** & Olsina, L. (2008), Assessing Web Applications Consistently: A Context Information Approach. En *Eighth International Conference on Web Engineering, ICWE '08.*, (IEEE Computer Society), págs. 224–230.
- Olsina, L.; Papa, F. & **Molina, H.** (2008), Ontological support for a measurement and evaluation framework. En *International Journal of Intelligent Systems*, (Wiley, New York, NY, USA), tomo 23(12); págs. 1282–1300, ISSN 0884-8173.
- Olsina, L.; **Molina, H.** & Papa, F. (2008), *Web Engineering: Modelling and Implementing Web Applications*, (Springer London), Capítulo 13. How to Measure and Evaluate Web Applications in a Consistent Way. ISBN 978-1-84628-922-4, págs. 385–420.
- **Molina, H.** & Olsina, L. (2007), Towards the Support of Contextual Information to a Measurement and Evaluation Framework. En *6th International Conference on Quality of Information and Communications Technology, QUATIC 2007*, (IEEE Computer Society), ISBN 0-7695-2948-8, págs. 154–166.
- **Molina, H.**, Olsina, L.; (2007) Soporte de Información Contextual en un Marco de Medición y Evaluación, Publicado en *Memorias del X Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes de Software (IDEAS'07)*, Isla de Margarita, Venezuela, 7-11 de Mayo. F. Losavio, G. H. Travassos, V. Pelechano, I. Diaz, A. Matteo (Eds.). ISBN: 978-980-325-323-3, 2007. págs. 77-90.
- Olsina, L., Papa, F. & **Molina, H.** (2005), Organization-Oriented Measurement and Evaluation Framework for Software and Web Engineering Projects. En *Proceedings of the International Conference on Web Engineering (ICWE) 2005*, tomo 3579 de *Lecture Notes in Computer Science* (Lowe, D. & Gaedke, M., eds.), (Springer-Verlag, Berlin Heidelberg), págs. 42–52.
- **Molina, H.**; Papa, F.; de los Angeles Martín, M. & Olsina, L. (2004), Semantic Capabilities for the Metrics and Indicators Cataloging Web System. En *Engineering Advanced Web Applications* (Matera, M. & Comai, S., eds.), (Rinton Press Inc.), ISBN 1-58949-046-0, págs. 97–109.

1.5. Estructura de la Tesis

El resto de esta tesis se organiza de la siguiente forma: en el Capítulo 2 se presenta el estado del arte relevante para este trabajo en cuanto a enfoques y propuestas de medición y evaluación de calidad de software así como a aspectos relacionados al enfoque sensible al contexto. En el Capítulo 3 se presentan los principios y fundamentos sobre los cuales se construye esta tesis, presentando primero, la estrategia de medición y evaluación utilizada, donde se describe el marco INCAMI y su base ontológica así como el proceso y la metodología asociada, y segundo el enfoque sensible al contexto utilizado, presentando una caracterización del término *contexto* tal cual se entiende en esta propuesta, los requerimientos deseables de la misma en cuando al manejo de información de contexto así como el enfoque de modelado y de representación elegido.

Luego, en la segunda parte se describe la propuesta central de esta tesis. En el Capítulo 4 se describen los módulos del marco conceptual C-INCAMI y sus interacciones prestando especial atención

a la especificación de la información de contexto. Además se describe el mecanismo de integración del dominio de aplicación al marco. En el Capítulo 5 se describen los aspectos del marco respecto del enfoque sensible al contexto, describiendo cómo se especifica la información de contexto en un proyecto de medición y evaluación así como para los elementos de información reusables durante la etapa de diseño, además del mecanismo diseñado para efectuar comparaciones de contexto como base para un sistema de recomendación. En el Capítulo 6 se presenta una *prueba de concepto* para mostrar la factibilidad de aplicar la propuesta en el diseño e implementación de un proyecto de medición y evaluación. En el Capítulo 7 se describe la arquitectura tecnológica diseñada para dar soporte a la propuesta y se presenta el prototipo que implementa sus componentes. Finalmente en el Capítulo 8 se presentan las conclusiones de este trabajo y posibles líneas futuras de investigación.

Estado del Arte



En este capítulo se presenta una revisión y una discusión crítica de la literatura relacionada a las áreas relevantes para el objetivo de esta tesis. Cabe destacar que en este trabajo se recurre a dos áreas de investigación en la disciplina de sistemas de información: la medición y evaluación de calidad de artefactos de software y las aplicaciones sensibles al contexto, hasta ahora no relacionadas o integradas en la literatura relacionada, como veremos a lo largo de este capítulo. Esta particularidad hace que la longitud del mismo se vea incrementada considerando trabajos de investigación en una u otra área. El capítulo en cuestión se estructura de la siguiente forma: en la Sección 2.1 se cubren aspectos de medición y evaluación de calidad en proyectos de software y web, a saber: *(i)* para destacar la importancia de contar con un mecanismo de medición y evaluación de entidades en proyectos de desarrollo de software, se verá el rol que cumplen los procesos de medición y evaluación dentro del ciclo de vida de desarrollo de software y sus interacciones y dependencias con los otros procesos; *(ii)* se elaborará sobre las particularidades relacionadas a la definición e implementación de la medición y evaluación de calidad de entidades y se verán los diferentes modelos que pueden utilizarse a tal fin; *(iii)* se verán diferentes enfoques que pueden ser utilizados para guiar la medición y evaluación, diferentes procesos que definen las actividades que deben llevarse a cabo para definir e implementar dichas actividades, así como los métodos que pueden aplicarse a lo largo de su implementación. Estos enfoques, procesos y métodos establecen un marco de referencia para la definición de modelos conceptuales de medición y evaluación; *(iv)* se identificarán los elementos que participan en las actividades de medición y evaluación en una primer instancia hacia la definición de modelos utilizados en la especificación de los datos y metadatos involucrados en dichas actividades; *(v)* se verán los marcos conceptuales para medición y evaluación utilizados para asistir en la definición e implementación de tales actividades; *(vi)* finalmente, se enfatizará sobre la evidencia existente en la literatura relacionada de la necesidad de contar con descripciones explícitas del contexto en el cual se llevan a cabo la medición y evaluación destacando la ausencia de marcos conceptuales que cubran este aspecto.

En la Sección 2.2 se cubren los diferentes aspectos del enfoque sensible al contexto asociados a su definición y su implementación: *(i)* se provee una descripción del enfoque y se analizan las definiciones propuestas por diferentes autores; *(ii)* se abarcan aspectos de representación de la información de contexto, analizando diferentes enfoques de modelado, así como aspectos de procesamiento de información de contexto considerando diferentes escenarios de uso; *(iii)* se analizan diferentes propuestas de aplicación del enfoque sensible al contexto a diferentes dominios de aplicación –tanto a dominios específicos como enfoques independientes del dominio–, y las ventajas y desventajas de cada propuesta.

Finalmente en la Sección 2.3 se elaboran conclusiones respecto de los trabajos relacionados a la luz del objetivo de esta tesis.

2.1. Medición y Evaluación de Calidad en Proyectos de Software y Web

2.1.1. Medición y evaluación como procesos de soporte

En el Capítulo 1 se dijo que la medición tiene como propósito conocer de forma objetiva el estado de las entidades relevantes en el desarrollo de software y atiende a las necesidades de comprender, evaluar, predecir y mejorar diversos aspectos de las mismas. Tales necesidades de información pueden ser requeridas en diferentes momentos del ciclo de vida de desarrollo de software, en el contexto de los diferentes procesos de desarrollo o gestión, tanto a nivel de proyecto como de la organización misma. Por esto, se considera a la Medición y Evaluación como procesos de soporte a los procesos de desarrollo y gestión, entre otros. Los procesos de soporte cubren actividades que apoyan y complementan los diferentes aspectos que rodean el desarrollo y mantenimiento de productos. Los procesos de Medición y Evaluación dan soporte a otros procesos al proveer actividades específicas que permiten alinear el estado de las entidades relevantes de proyectos y productos con los objetivos y necesidades de información gerenciales durante la ejecución de todo el ciclo de vida de desarrollo. Los resultados arrojados por estas actividades pueden ser usados en la toma de decisiones y en la ejecución de acciones correctivas apropiadas respecto del estado de tales entidades.

El rol de la Medición y Evaluación como proceso de soporte son cubiertos en más o menos detalle por diferentes autores en la literatura relacionada. En [Pressman, 2001] se provee una vista genérica de la Ingeniería de Software en la que las actividades pueden ser categorizadas en tres fases genéricas: definición, desarrollo y soporte (éste último se enfoca a los cambios en los productos asociados a correcciones, adaptaciones, mejoras o prevención). Estas fases son complementadas por un número de *actividades paraguas*, independientes a las actividades principales de la Ingeniería de Software y comunes a todo proyecto de software, que se superponen al modelo de proceso. Entre estas *actividades paraguas* el autor incluye la medición –entre otras como aseguramiento de calidad, gestión de la configuración, gestión de riesgos, etc. La integración de la medición al ciclo de vida del software también es mostrada por [Zuse, 1998] donde se discuten un conjunto de posibles métricas de producto y de proceso para las diferentes fases de un ciclo de vida del software simple, que involucra las fases de especificación, diseño, codificación, prueba y mantenimiento.

Por otro lado, el modelo *CMMI® for Development v1.2* [CMMI Product Team, 2006] incluye, entre sus áreas de proceso de soporte, a *Medición y Análisis*, que es utilizada por todas las áreas de proceso restantes (ver Figura 2.1), inclusive por las actividades definidas en las prácticas genéricas. Además provee información más detallada acerca de la forma en que las prácticas de *Medición y Análisis* pueden ser utilizadas en cada una de las áreas de proceso descriptas. Las dependencias directas que surgen de este detalle pueden observarse en la Tabla 2.1. Otras áreas de proceso dependen de *Medición y Análisis* de forma indirecta, es decir, dependiendo de alguna de las áreas de proceso que se mencionan en dicha tabla.

Otras fuentes de la literatura relacionada describen las interacciones de los procesos de la Ingeniería de Software con la M&E sin definir un ciclo de vida como es el caso del *Guide to the Software Engineering Body of Knowledge (SWEBOK)* [Abran et al., 2004] y del *Project Management Body of Knowledge (PMBOK)* [PMI, 2000]. Ambos documentos contienen un compendio del cuerpo de conocimiento que existe en la literatura publicada en las últimas décadas para ambas disciplinas. Estos documentos se encuentran estructurados en *Áreas de Conocimiento* que a su vez se encuentran subdivididas en tópicos que cubren diferentes aspectos de cada área.

En SWEBOK la medición aparece, por un lado, como una de las actividades de gestión de la Ingeniería de Software, describiendo de forma genérica el proceso de medición definido en el estándar ISO 15939 [ISO/IEC 15939, 2001], y por otro lado refiriéndose a la medición de procesos y productos como uno de los tópicos que componen el proceso de la Ingeniería de Software, destacando su rol en

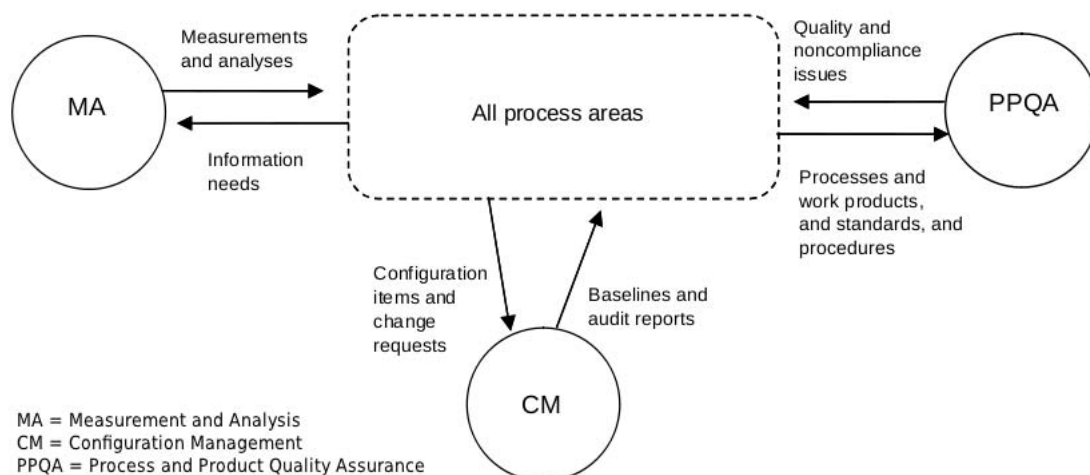


Figura 2.1: Áreas de proceso de soporte básicas.

<i>Área de Proceso</i>	<i>Categoría</i>	<i>Nivel</i>
Monitoreo y Control de Proyecto	gestión de proyecto	2
Verificación	ingeniería	3
Desarrollo de Requerimientos	ingeniería	3
Gestión de Proyecto Integrado + IPPD	gestión de proyecto	3
Foco de Proceso Organizacional	gestión de proceso	3
Definición de Proceso Organizacional + IPPD	gestión de proceso	3
Gestión de Proyecto Cuantitativo	gestión de proyecto	4
Desempeño de Proceso Organizacional	gestión de proceso	4
Innovación y Desarrollo Organizacional	gestión de proceso	5
Análisis y Resolución Causal	soporte	5

Tabla 2.1: Áreas de Proceso de CMMI que dependen directamente del área de proceso Medición y Análisis.

la mejora de procesos y productos y cubriendo otros aspectos relevantes tales como: la calidad de las mediciones –esenciales para proveer resultados efectivos–; el uso de modelos de información de calidad con fines de análisis, clasificación y predicción; y técnicas de medición de procesos, tanto analíticas como comparativas.

Otra de las fuentes que evidencian la integración de la medición y evaluación al ciclo de vida de desarrollo de software es el estándar ISO/IEC 14598 [ISO/IEC 14598-1, 1999] donde se muestran las diferentes mediciones y evaluaciones que se llevan a cabo para cada una de las etapas del ciclo, tal como se muestra en la Figura 2.2.

La importancia de contar con una capacidad de medición y evaluación en las distintas etapas del ciclo de vida de desarrollo es hecha explícita en [Munson, 2003] como un requisito necesario. Durante la etapa de requerimientos la medición permite obtener indicadores de los costos de desarrollo, durante el diseño permite la evaluación de alternativas de diseño, durante las actividades de prueba provee datos sobre la adecuación de las técnicas de prueba y sobre el tiempo esperado para producir un producto que cumpla un cierto nivel de calidad. Además, afirma que un buen programa de medición permite la óptima asignación de recursos limitados, como son las horas y capacidades del equipo de desarrollo, recursos de hardware, entre otros, para producir el mejor resultado posible en el menor tiempo posible y con los recursos disponibles. Agrega que la contribución más importante de un programa de medición es el establecimiento de una base histórica de experiencias, que puede ser utilizada para proveer buenos estimadores de costos, calidad y confiabilidad para proyectos futuros.

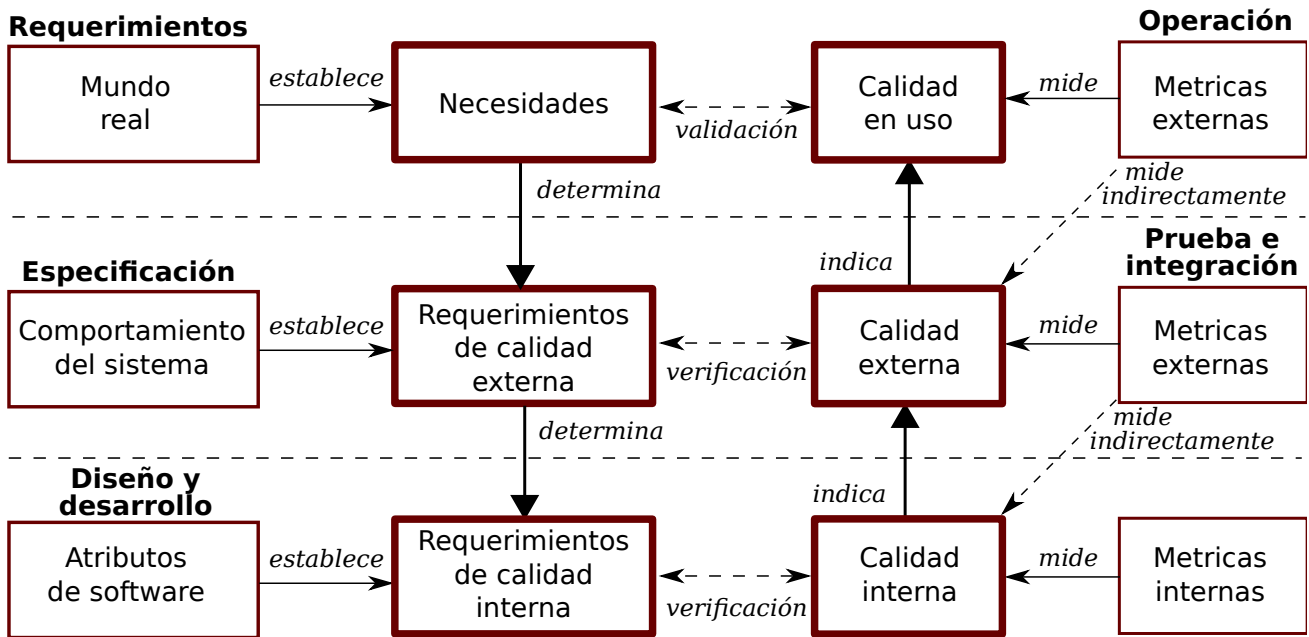


Figura 2.2: Integración del proceso de evaluación al ciclo de vida de software según el estándar ISO/IEC 14598-1:1999 (adaptada de dicho estándar).

Como se vio en los trabajos citados arriba los procesos de medición y evaluación son incluidos a lo largo de todo el ciclo de vida de desarrollo de software, y son utilizados por todos los otros procesos de forma directa o indirecta, para proveer un mecanismo que permite conocer el estado relevante de cada entidad participante en el mismo. Las mediciones son utilizadas para controlar la gestión de proyectos asociada a riesgos, costos, calidad, etc., la mejora de procesos, el desarrollo de requerimientos, el diseño de la solución, la calidad de los productos desarrollados y el mantenimiento de los mismos, etc. Este mecanismo es la fuente fundamental de información para la toma de decisiones que permite controlar y efectuar las correcciones necesarias para satisfacer los objetivos establecidos, tanto a nivel de proyecto como a nivel organizacional. Desde la definición y planificación de un proyecto se utilizan mediciones para establecer los parámetros actuales y deseables de las entidades del proyecto, en base a mediciones históricas. A lo largo de su ejecución, la medición permite conocer el estado actual que será comparado con los valores establecidos en el comienzo, descubrir desviaciones de los mismos y posibles causas, todo esto de forma cuantitativa, lo que permite también corregir que éstos vuelvan a los valores previstos. De allí la importancia de contar con programas de medición y evaluación robustos y claramente definidos que permitan conocer de forma eficiente y coherente el estado relevante de cada entidad en proyectos de desarrollo de software.

2.1.2. Medición y evaluación de calidad

En la subsección anterior se describió a *calidad* como un aspecto clave en la medición y evaluación de productos de software. Para poder medir y evaluar la *calidad* de cualquier artefacto de software es necesario primero entender el concepto mismo de *calidad*, lo que permitirá especificarlo de forma clara y estructurada.

En la literatura existen diversas definiciones del término. En el estándar ISO 9126 [ISO/IEC 9126-1, 2001], el término *calidad de software* se define como “la totalidad de las características de un producto de software que influyen en su habilidad de satisfacer necesidades explícitas o implícitas” y el término *características de calidad de software* es descrito como “un conjunto de atributos de un producto de software por el cual su calidad es descripta y evaluada”.

Así mismo, en el estándar sucesor SQuARE, donde se definen modelos de calidad y calidad en uso

[ISO/IEC 25010, 2011], se distinguen tres perspectivas de calidad para cada una de los cuales provee un modelo: (i) calidad de software, (ii) calidad en uso y (iii) calidad de datos. El término *calidad de software* se define como “el grado en que el producto de software satisface necesidades expresadas o implícitas cuando se usa bajo condiciones específicas.” En esta perspectiva se distingue entre *calidad externa* e *interna*: la *calidad externa* provee una vista de caja negra del producto de software y cubre propiedades relacionadas al producto en ejecución, mientras que la *calidad interna* provee una vista de caja blanca del producto y cubre propiedades estáticas del mismo que se encuentran comúnmente disponibles durante el desarrollo. La calidad interna tiene impacto en la calidad externa y ésta última tiene impacto a su vez sobre la calidad en uso. La segunda perspectiva, *calidad en uso*, se define como el grado en el cual un producto de software usado por usuarios específicos cubre sus necesidades de alcanzar objetivos o realizar tareas específicos con efectividad, eficiencia, flexibilidad, seguridad y satisfacción en contextos de uso específicos, particularmente, el entorno de operación de la aplicación, ya sea real o simulado. La calidad en uso es afectada por la calidad del software, hardware, sistema operativo, las características de los usuarios, las tareas y el ambiente social. Por último, *calidad de datos* es definida en el estándar ISO/IEC 25012 [ISO/IEC 25012, 2008] como “el grado en el cual las características de los datos satisfacen necesidades enunciadas e implicadas cuando son usados bajo condiciones especificadas”. Así como para calidad de software, se distinguen dos puntos de vista para calidad de datos: *calidad de datos inherente*, se refiere a las características de calidad que tienen el potencial intrínseco de satisfacer necesidades enunciadas e implicadas cuando son usados bajo condiciones especificadas; y *calidad de datos dependiente del sistema*, se refiere al grado en que las características de calidad son satisfechas y preservadas en un sistema de cómputos cuando los datos son usados bajo condiciones especificadas.

En [Lew et al., 2010] se propone extender la perspectiva de *calidad de software* del estándar ISO/IEC 25010 (aunque de una versión anterior a la citada anteriormente¹) introduciendo la característica *calidad de información* que los autores definen como “el grado en que el software/aplicación web provee información precisa, apropiada, accesible y legalmente conforme”, así como también la perspectiva de *calidad en uso* introduciendo la característica *capacidad de aprendizaje en uso*² definida por los autores como “el grado en que usuarios especificados pueden aprender eficientemente y efectivamente alcanzando objetivos especificados en contextos de uso especificados”. Además los autores proponen una nueva perspectiva, *experiencia real de usuario* definida como “el grado en que usuarios especificados pueden alcanzar usabilidad, seguridad y satisfacción en uso reales en un contexto de uso especificado”, pensada para evaluar características particulares de las aplicaciones web. Esta propuesta tiene como objetivo cubrir la falta de modelos para evaluar los aspectos mencionados de las aplicaciones web –no incluidos en el estándar ISO 25010, cada vez más importantes debido a su evolución como forma de implementación predominante. Los modelos correspondientes a estas perspectivas son revisados en la Sección 2.1.5.

Juran [Juran & Godfrey, 1999] contempla dos significados para el término calidad: 1) la existencia de características en el producto que satisfacen las necesidades de los usuarios/clientes y 2) la ausencia de deficiencias que pudieran requerir retrabajo o causar fallas en el producto y, por lo tanto, la insatisfacción del cliente.

Pressman [Pressman, 2001] también considera dos significados para el término calidad, basados en la satisfacción de especificaciones que responden a dos grupos: *calidad de diseño*, se refiere a las características que los diseñadores especifican para un producto respecto de los materiales, tolerancias y desempeño esperados, y *calidad de conformidad*, se refiere a las especificaciones de diseño a ser seguidas durante el desarrollo.

En [Kan, 2002] se plantea la dificultad de proveer una definición de calidad debido a que es un concepto multidimensional que depende de la entidad de interés, el punto de vista para esa entidad y los atributos de calidad de la misma, y a que su definición acepta diferentes niveles de abstracción y/o

¹ISO/IEC CD 25010.3 (2009)

²Del inglés “*learnability in use*”.

detalle. El autor también presenta dos visiones de calidad: por un lado interpreta el concepto *calidad* como la *conformidad con un conjunto de requerimientos* que deben ser claramente especificados para la entidad, y por otro lado como *adecuación para su uso* basado en requerimientos y expectativas de los clientes. En conjunto, estas vistas responden en general a una interpretación de calidad como *conformidad con requerimientos de clientes* y puede definirse, de forma operacional, en dos niveles: calidad intrínseca del producto y satisfacción del cliente.

De las definiciones e interpretaciones anteriores, se puede extraer una caracterización del término *calidad*, a saber:

- es una cualidad o característica atribuible a las entidades, desde algún punto de vista y en un contexto particular,
- responde a un conjunto de propiedades deseables para la entidad, que satisfacen las necesidades implícitas o explícitas para el punto de vista en consideración, que provienen de:
 - características inherentes al producto respecto de los materiales, tolerancias y desempeño esperados (confiabilidad, eficiencia, rendimiento),
 - características asociadas al proceso de desarrollo del producto (modificabilidad, modularidad),
 - necesidades específicas de los usuarios/clientes,
 - características deseables en la interacción del usuario con el producto en un contexto de uso (usabilidad, efectividad, satisfacción),
- la medición y evaluación de calidad tiene como objetivo determinar el grado o nivel en el que una entidad satisface un conjunto especificado de requerimientos en un contexto particular, por lo que cuanto mayor cantidad de requerimientos implícitos sean explicitados, mayor será la precisión de la evaluación.

Ya que la calidad es un concepto complejo que no puede ser cuantificado de forma directa, el nivel de satisfacción de un conjunto de requerimientos puede ser calculado a partir de los valores de las propiedades o *atributos* que corresponden al mundo empírico y que caracterizan a las entidades. Para poder utilizar los atributos de una entidad en un valuación de calidad se debe realizar una correspondencia o conversión del mundo empírico al mundo formal o matemático. La definición de esta correspondencia es llamada *métrica* e incluye la especificación del dominio (el mundo empírico) y el rango (el mundo formal) así como las reglas para realizar la conversión. El rango es especificado mediante una escala de algún tipo (nominal, ordinal, intervalo, proporción y absoluta) pudiendo ser numérica o simbólica; su elección depende del tipo de atributo a representar más apropiada. Para poder ser utilizada en la valuación de calidad de una entidad, toda métrica debe ser validada [ISO/IEC 14598-1, 1999]: los valores del mundo formal deben preservar las relaciones existentes entre los valores del mundo real (*condición de representación*); de otra forma, los resultados obtenidos a partir de una métrica no serían coherentes con el mundo real. La ejecución efectiva de una métrica es llamada *medición*, y el resultado producido por la misma es una *medida*, que consiste en un número o símbolo que corresponde al valor del atributo en el mundo formal. En muchos casos los atributos pueden ser medidos directamente de la entidad correspondiente, constituyendo el bloque de construcción de la medición. Atributos de este tipo son, por ejemplo, *tamaño de código fuente*, *duración del proceso de prueba*, *número de defectos descubiertos* y *tiempo invertido en un proyecto*. Sin embargo, en muchos casos los atributos definidos para una entidad dada deben ser calculados a partir de otros atributos de la misma entidad (u otras entidades relacionadas) mediante alguna fórmula. Éstas mediciones son más útiles en hacer visibles las interacciones entre atributos medidos de forma directa y por consiguiente hacer evidente lo que está sucediendo en un proyecto.

Otro de los aspectos a tener en cuenta en la definición de métricas es el de tratar de mantener las mediciones objetivas de manera que las mismas sean repetibles y consistentes cuando se midan productos, procesos o recursos. Por otro lado, las métricas subjetivas dependen del contexto para el cual fueron hechas y en el cual son aplicadas [Fenton & Pfleeger, 1997]. Las medidas obtenidas de estas métricas pueden variar con la persona que las realiza, reflejando el cambio de juicio del medidor, y puede ser difícil llegar a un consenso en los valores medidos. En lo posible debe reducirse al máximo la subjetividad de una métrica; en caso contrario debieran registrarse los factores que la hacen subjetiva para poder utilizar e interpretar sus valores de manera informada y coherente. No obstante, las medidas subjetivas son particularmente importantes porque pueden ser útiles en casos donde se requiera de un juicio subjetivo para la valuación de un atributo. Por ejemplo, para medir la calidad de los requerimientos antes de continuar con el desarrollo de los mismos se podría utilizar una métrica que registre el nivel de comprensión de los requerimientos por parte de los desarrolladores en una escala ordinal numérica de 1 a 5 (del mejor al peor caso). En este ejemplo, la subjetividad de la métrica es deseable ya que refleja efectivamente la habilidad del equipo de trabajo para continuar con el proceso de desarrollo.

De la misma forma que dos o más atributos son combinados para calcular o medir de forma indirecta otro atributo, el concepto de *calidad* puede ser estimado combinando atributos de la entidad a medir mediante un *modelo de calidad*. Un modelo de calidad incluye un conjunto de características y atributos de la entidad a medir y las relaciones entre ellos. Así, el modelo constituye una parte esencial y la base para especificar requerimientos de calidad para evaluar la satisfacción de los objetivos de calidad [ISO/IEC 14598-1, 1999]. Esta evaluación se lleva a cabo midiendo los atributos incluidos en el modelo y luego interpretando las subcaracterísticas y características para determinar el grado o nivel en que éstas están presentes en el producto. Existen diferentes tipos de modelos: de texto, esquemáticos y algorítmicos [Laird & Brennan, 2006]. En los *modelos de texto* se utiliza lenguaje natural para describir las características deseables en la entidad a medir. Estos modelos no son efectivos para describir situaciones complejas pero pueden ser el punto de partida para comprender los requerimientos de calidad. Los *modelos esquemáticos* permiten modelar las entidades, las relaciones entre ellos y su dinámica utilizando una notación gráfica apropiada, por lo que pueden ser muy poderosos. Por último, los *modelos algorítmicos*, también llamados *paramétricos*, permiten representar claramente las relaciones entre los diferentes requerimientos de una entidad mediante una función o expresión matemática, permitiendo realizar una estimación cuantitativa de calidad; por esta razón pueden ser extremadamente poderosos. También es posible una combinación de diferentes tipos de modelos para complementar sus capacidades, siempre y cuando sean compatibles.

Los modelos de calidad también pueden clasificarse según su flexibilidad para ser adaptados a las necesidades particulares [Fenton & Pfleeger, 1997]:

- Modelos fijos: se asume que todos los factores de calidad necesarios son un subconjunto de los propuestos por el modelo y se aceptan las relaciones entre factores, criterios y métricas de calidad definidos por el modelo.
- Modelos *ad hoc* o propios: se acepta la filosofía general que calidad está compuesta por muchos atributos pero no se adopta la caracterización de calidad de un modelo dado. En su lugar se determinan, por consenso con potenciales usuarios o expertos, los atributos de calidad importantes para un producto dado así como su descomposición en atributos de mas bajo nivel y las relaciones entre ellos (posiblemente guiado por un modelo existente).
- Modelo mixto: se parte de un modelo de calidad fijo y se modifica apropiadamente para satisfacer necesidades particulares como en los modelos propios.

En la literatura existen diferentes modelos de calidad propuestos, entre los cuales se destacan los de McCall y Bohem [Berander *et al.*, 2005; Fenton & Pfleeger, 1997] como predecesores de los modelos actuales. Ambos modelos se enfocan en el producto final e intentan definir cuantitativamente la calidad

de software mediante un conjunto fijo de características organizadas jerárquicamente en diferentes niveles de abstracción hasta el nivel más bajo, directamente cuantificable, cada una de las cuales contribuye al nivel de calidad general.

El modelo de McCall (ver Figura 2.3) se organiza en tres perspectivas para identificar y definir la calidad de un producto de software: *revisión del producto* (habilidad para efectuar cambios), *transición del producto* (adaptabilidad a nuevos ambientes) y *operación del producto* (características relacionadas al uso del producto). Para cada perspectiva se define un conjunto de **factores** que describen diferentes características de comportamiento general del sistema; los factores son descompuestos en **critérios** que a su vez son descompuestos en **métricas**, que proveen una escala y método para efectuar la medición directamente sobre el producto. Para obtener una vista completa de la calidad de un producto utilizando este modelo, se sintetizan las métricas para cada criterio y luego para cada factor. Las métricas se basan en responder preguntas del estilo *si/no* y se sintetizan contabilizando cada respuesta diferente como un porcentaje del total de métricas para cada criterio. Los factores se sintetizan de la misma forma a partir de los valores resultantes de los criterios.

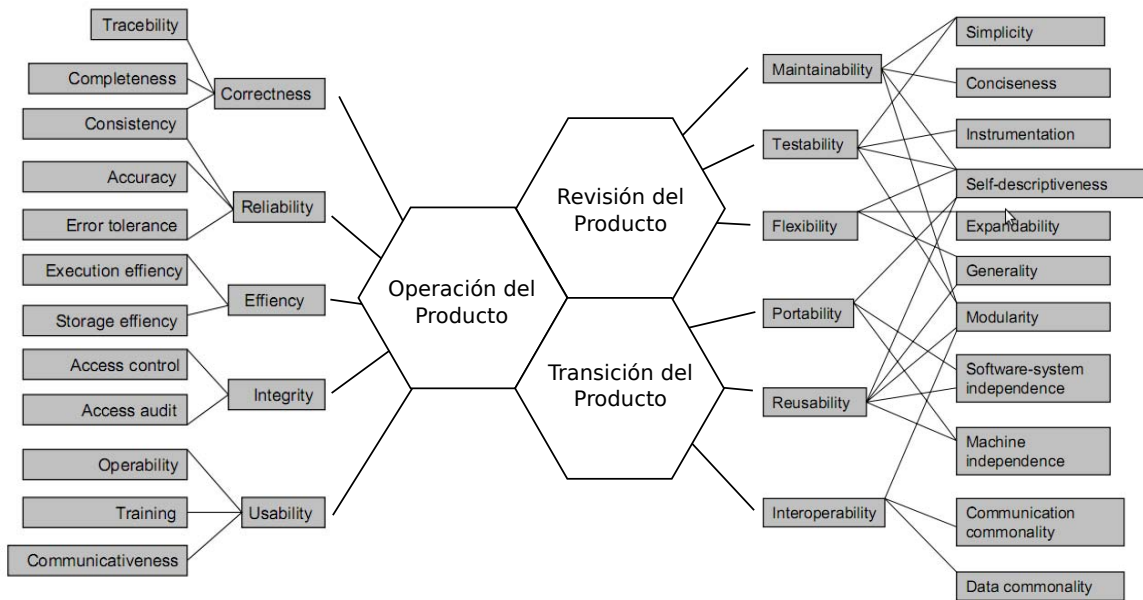


Figura 2.3: Modelo de Calidad de McCall.

Por otro lado, el modelo de Boehm (ver Figura 2.4) incluye **características de alto nivel** que representan los requerimientos básicos del uso real del producto, es decir, la utilidad general del software. Estas cubren tres preguntas principales: *utilidad* (facilidad, confiabilidad, eficiencia), *mantenibilidad* (facilidad para entender, modificar y probar) y *portabilidad* (capacidad de utilizar del producto en diferentes entornos). Las características del nivel intermedio representan los **factores de calidad** que en conjunto representan la calidad esperada en un sistema de software. El nivel más bajo de la jerarquía corresponde a las **características primitivas**. Estas proveen la base para definir métricas³ de calidad, pudiendo utilizar más de una para medir una determinada característica primitiva.

Otro de los modelos de calidad que se destaca en la literatura es el definido en el estándar ISO/IEC 9126-1 [ISO/IEC 9126-1, 2001], basado en el modelo de calidad de McCall, destinado para la evaluación de cualquier tipo de software. El estándar provee tanto un modelo para calidad interna y externa así como para calidad en uso (ver Figura 2.5). El modelo para calidad interna y externa está compuesto de seis características de calidad de producto, cada una de las cuales está refinada en un conjunto de subcaracterísticas, y cada una de estas, a su vez, son evaluadas por un conjunto de métricas que

³Definido por Boehm como: "a measure of extent or degree to which a product possesses and exhibits a certain (quality) characteristic".

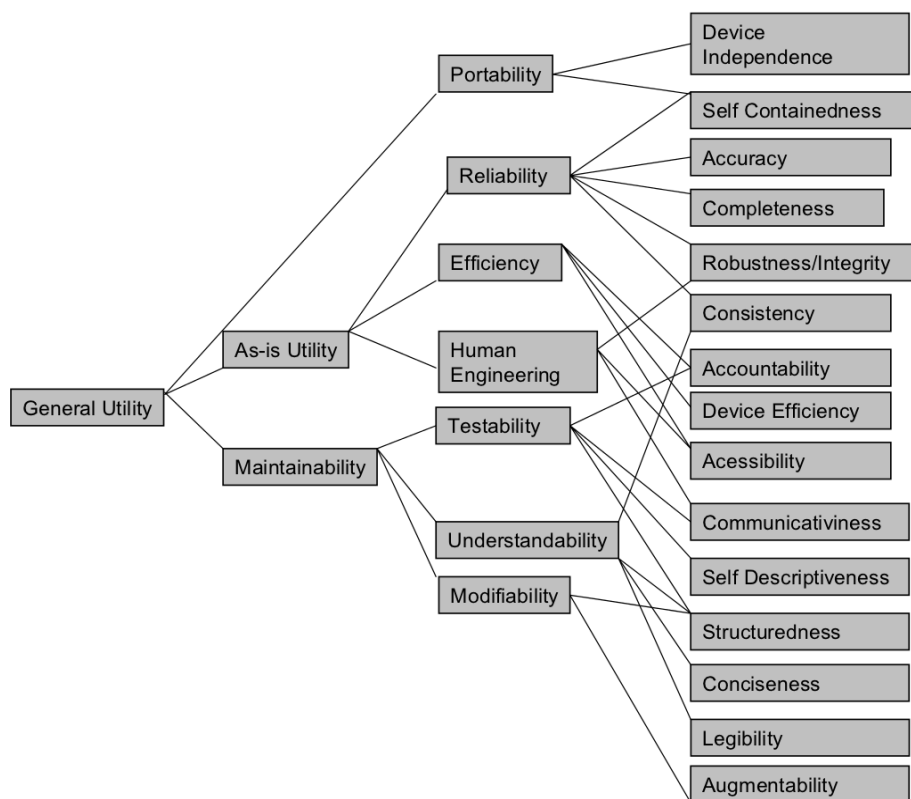


Figura 2.4: Modelo de Calidad de Boehm.

cuantifican atributos de la entidad a medir. El modelo propuesto no define los atributos en que se descompone cada subcaracterística, ya que varían entre diferentes productos software; sin embargo el estándar provee de manera informativa un conjunto de métricas a utilizar para medir atributos internos y externos de producto. A diferencia del modelo de calidad interna y externa, el modelo de calidad en uso solo provee un solo nivel de características. De esta forma, el estándar promueve que las organizaciones definan su propio modelo de calidad a partir del modelo de propuesto (modelo de tipo mixto). En este documento, a diferencia de otros autores vistos anteriormente, se utiliza el término “métrica” para referirse a la especificación del método utilizado para realizar una medición junto a una escala cuantitativa utilizada para representar sus valores, mientras que el término “medida” es usado para referirse al resultado de la medición.

Como una mejora al modelo de calidad del estándar ISO/IEC 9126-1, se crea la colección de estándares ISO/IEC 2501n que constituye la *División de Modelos de Calidad* de la familia de estándares ISO/IEC 25000 (SQuaRE). Los estándares de ésta división presentan modelos detallados de calidad de software, calidad en uso y calidad de datos, así como guías prácticas para su aplicación. Los modelos de calidad de SQuaRE [ISO/IEC 25010, 2011] clasifican la calidad de producto en características que pueden subdividirse en subcaracterísticas que las influyen. Tanto las características como las subcaracterísticas pueden ser cuantificadas por medio de propiedades de calidad, cada una con métricas⁴ de calidad asociadas. No obstante, las características y subcaracterísticas definidas en el estándar pueden ser usadas como una lista de comprobación para asegurar una amplia cobertura de requerimientos y se permite la adición de nuevos niveles de subcaracterísticas en la estructura jerárquica. Éste estándar incluye dos modelos de calidad: el modelo para calidad externa e interna y el modelo para calidad en uso. El modelo de calidad externa e interna presentado en este estándar (ver Figura 2.6), a diferencia del presentado en ISO/IEC 9126-1, define ocho características de calidad de software –*adecuación funcional, eficiencia del desempeño, compatibilidad, usabilidad, confiabilidad, seguridad, mantenibilidad y portabilidad*, y subcaracterísticas asociadas. De la misma forma, el modelo de calidad en uso presentado

⁴El estándar ISO/IEC 25010 se refiere a éstas como “medidas”.

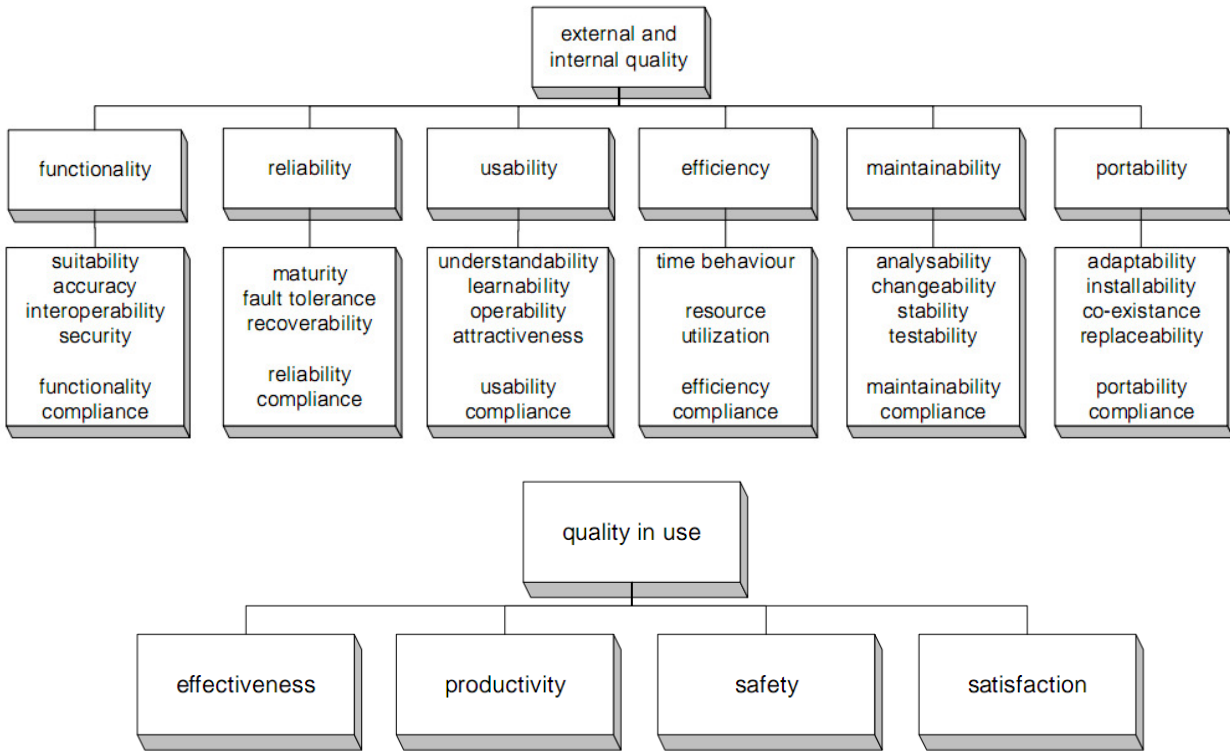


Figura 2.5: Modelo de calidad y calidad en uso del ISO 9126-1.

(ver Figura 2.7) incluye cambios adicionales. Se compone de cinco características relacionadas al uso de un sistema –*efectividad, eficiencia, satisfacción, libertad de riesgo y cobertura del contexto*, algunas de ellas compuestas de un conjunto de subcaracterísticas.

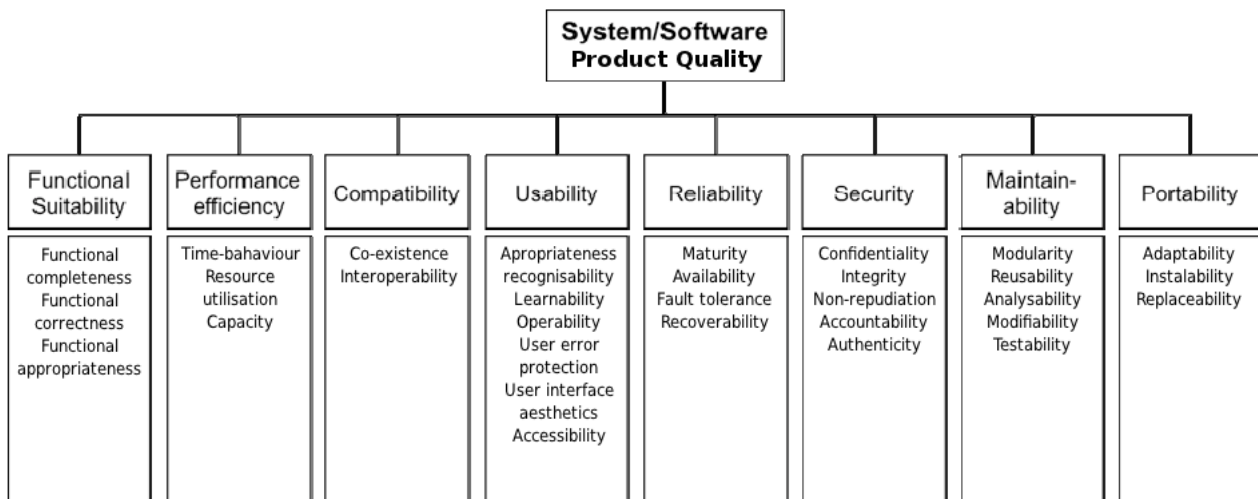


Figura 2.6: Modelo de calidad de producto del estándar ISO/IEC 25010 (tomado de [ISO/IEC 25010, 2011]).

Adicionalmente SQuaRE presenta un modelo de calidad para datos estructurados manipulados por sistemas de cómputos [ISO/IEC 25012, 2008]. Este modelo complementa a los definidos en el estándar ISO/IEC 25010 y define características o requerimientos de calidad para cualquier dominio de aplicación. El modelo de calidad de datos (presentado en la Figura 2.8) está estructurado en quince características consideradas desde dos puntos de vista: inherentes e independiente del sistema. Las características de calidad inherentes a los datos se refieren al potencial intrínseco de los mismos de satisfacer necesidades

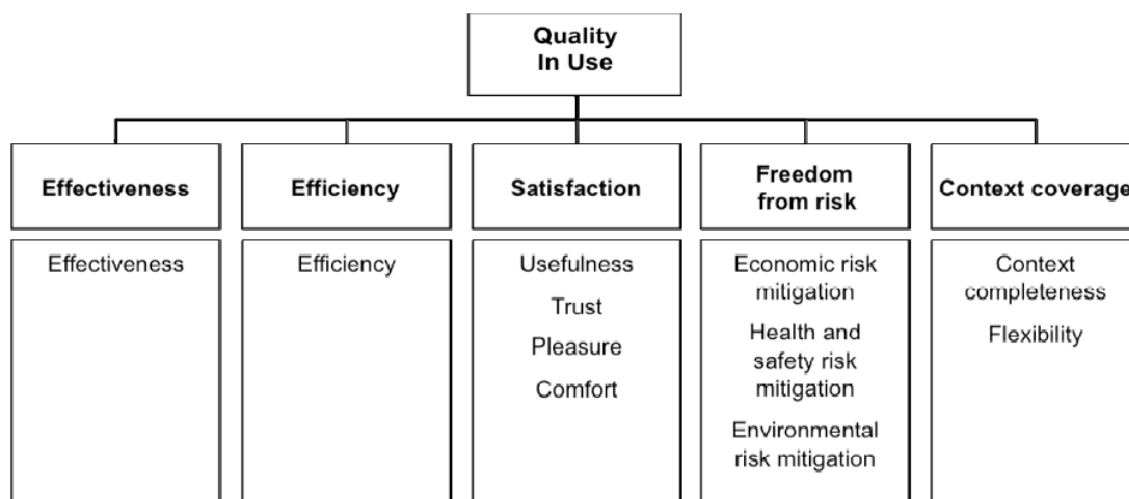


Figura 2.7: Modelo de calidad en uso del estándar ISO/IEC 25010 (tomado de [ISO/IEC 25010, 2011]).

enunciadas e implícitas cuando son usados bajo condiciones especificadas. Estas características apuntan a los valores del dominio de datos y posibles restricciones, relaciones en los valores de datos y a los metadatos correspondiente. Por otro lado, las características dependientes del sistema se refieren al grado en que la calidad de datos es alcanzada y preservada en un sistema de cómputos cuando los datos son usados bajo condiciones especificadas. Desde este punto de vista la calidad de los datos depende del dominio tecnológico en que los datos son usados y es alcanzada por las capacidades de los componentes de los sistemas de cómputos tales como: dispositivos de hardware o componentes de software del sistema de cómputos (por ejemplo para proveer facilidades de precisión, recuperabilidad o portabilidad). Todas las características del modelo son definidas y se proveen, para cada una, ejemplos de métricas que pueden cuantificarlas.

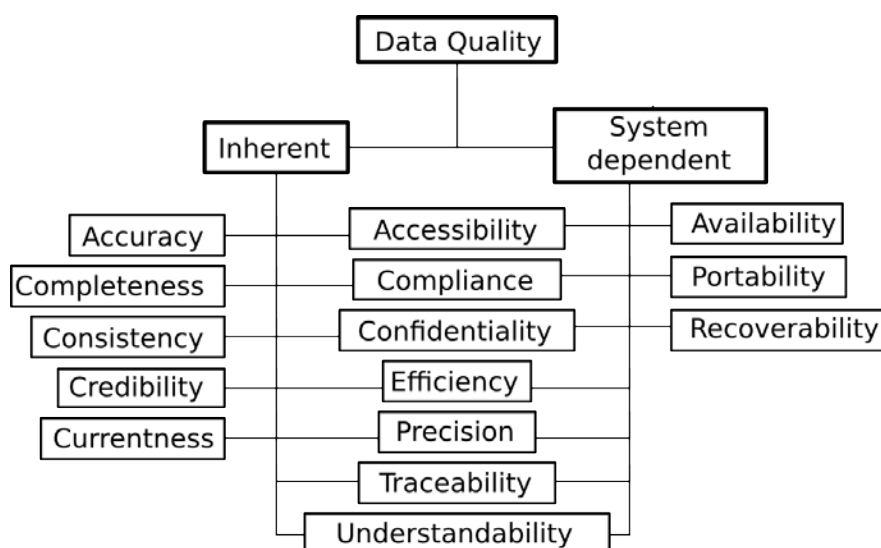


Figura 2.8: Modelo de calidad de datos del estándar ISO/IEC 25012 (adaptado de [ISO/IEC 25012, 2008]).

Todos los modelos de SQuaRE pretenden ser una guía para la definición de requerimientos de calidad pudiendo ser adaptados en función del tipo de producto, el dominio de aplicación, los objetivos del negocio, la naturaleza del producto y el proceso de desarrollo.

En [Lew *et al.*, 2010] se propone una extensión a los modelos de calidad interna/externa y calidad en

uso definidos en el estándar ISO/IEC 25010⁵ definiendo e incorporando nuevas características, críticas para la evaluación de aplicaciones web; además proponen un nuevo modelo para evaluar características particulares de las aplicaciones web, no incluida en los modelos del estándar mencionados. Estas nuevas características fueron introducidas en la sección 2.1.2. Los autores incorporan al modelo de calidad interna/externa la característica *calidad de información* que incluye las subcaracterísticas *precisión, adecuación, accesibilidad y cumplimiento legal*. Por otro lado, el modelo de calidad en uso es extendido agregando la subcaracterística *capacidad de aprendizaje en uso a usabilidad en uso*. La nueva perspectiva propuesta por los autores, *experiencia real de usuario* (ver Figura 2.9) es modelada en función de un conjunto de características y subcaracterísticas del modelo de calidad en uso del estándar ISO 25010 , más una nueva característica, *usabilidad real*, que los autores definen como “el grado en que usuarios especificados pueden alcanzar objetivos especificados con *efectividad en uso, eficiencia en uso, capacidad de aprendizaje en uso y accesibilidad en uso*, en un contexto de uso especificado” aclarando que debe ser medida y evaluada en entornos operacionales reales donde los usuarios reales lleven a cabo tareas reales. Los autores instancian además los modelos propuestos con los atributos necesarios para realizar la evaluación correspondiente en aplicaciones web reales.

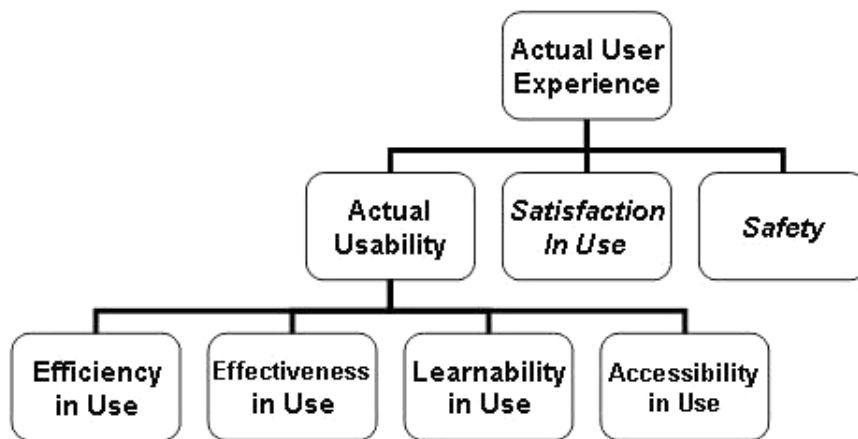


Figura 2.9: Modelo de Experiencia Real de Usuario de Lew et al.

En [Seffah et al., 2006] se presenta un modelo de calidad en uso que incluye aspectos relevantes a la perspectiva del usuario final, descartando aspectos internos del producto. El modelo sigue una estructura jerárquica como en los modelos de Bohem, McCall y los estándares ISO/IEC vistos anteriormente, aunque distingue cuatro niveles, a saber: factores, criterios, métricas⁶ y datos. Las métricas pueden utilizar dos posibles métodos para obtener un resultado: mediante conteo o cálculo. Los datos resumen el estado del criterio correspondiente pudiendo ser cuantitativos o cualitativos. El modelo propone factores como efectividad, eficiencia, satisfacción, productividad, seguridad, entre otros.

Otros autores proponen enfoques o modelos genéricos destinados a la definición de modelos de calidad adaptables a las necesidades particulares. Estas propuestas describen la estructura utilizada para relacionar los diferentes componentes del modelo y posiblemente un conjunto posible de factores o atributos de calidad para utilizar en el mismo.

Un ejemplo de estos modelos es el propuesto por Gilb en el que los requerimientos de calidad y de recursos son especificados individualmente para cada sistema de software –incluyendo una indicación de la prioridad relativa de los mismos, en función de un conjunto de atributos de calidad de alto nivel [Kitchenham, 1987]. Los atributos de calidad son subdivididos en atributos más específicos, hasta que se alcance un nivel en el cual el atributo puede ser medido directamente. Adicionalmente, Gilb

⁵ISO/IEC CD 25010.3 (2009), anterior a la citada en la sección 2.1.2

⁶Definido como “una función cuyas entradas son datos de software y cuya salida es un valor numérico único que puede ser interpretado como el grado en el cual el software posee un atributo dado que afecta su calidad”.

propone un conjunto de atributos de calidad de alto nivel –trabajabilidad, disponibilidad, adaptabilidad y usabilidad- junto a un conjunto de atributos para recursos –tiempo, dinero, gente y herramientas. Gilb también sugiere un rango de métricas ⁷ para cuantificar los atributos propuestos. Las métricas sugeridas utilizan una plantilla para su definición y están pensadas para ser definidas de forma local y no necesariamente transferibles. Este enfoque es fuertemente dependiente de la aplicación, impidiendo la comparación entre proyectos.

Otra de las propuestas que siguen este enfoque es el modelo constructivo de calidad, COQUAMO, de Kitchenham [Kitchenham, 1987]. COQUAMO es un modelo de calidad general para productos, basado en atributos de la organización, personal, producto, proyecto o proceso que influyen la calidad del producto, que necesita ser calibrado al ambiente particular de aplicación. El modelo tiene tres objetivos:

- predecir la calidad del producto final, considerando factores de calidad comunes a la mayoría de las aplicaciones o independiente de la aplicación,
- monitorizar el progreso de la calidad de producto siguiendo un conjunto de guías, y
- evaluar la calidad del producto final para validar las predicciones hechas y mejorar las futuras.

COQUAMO está basado en las propuestas de Gilb, McCall y otros, pero modifica el modelo simple de calidad considerando conceptos del ciclo de vida y de aseguramiento de calidad; además clasifica los factores de calidad identificando factores de calidad generales y medibles de forma directa en el producto final, como base para un modelo de calidad general, así como factores de calidad que pueden ser monitorizados durante el desarrollo, sugiriendo criterios y métricas para los mismos. A diferencia de los modelos de McCall, Bohem y Gilb, el modelo de COQUAMO descompone los factores de calidad en un conjunto de listas de comprobación que reflejan las distintas etapas del proceso de desarrollo de software, cada una de estas involucrando un conjunto de procedimientos, estándares y métricas (ver Figura 2.10). La satisfacción de cada factor de calidad puede ser juzgada contra su especificación utilizando estas comprobaciones de aceptación en cada etapa del ciclo de vida para monitorizar el progreso y al final del desarrollo.

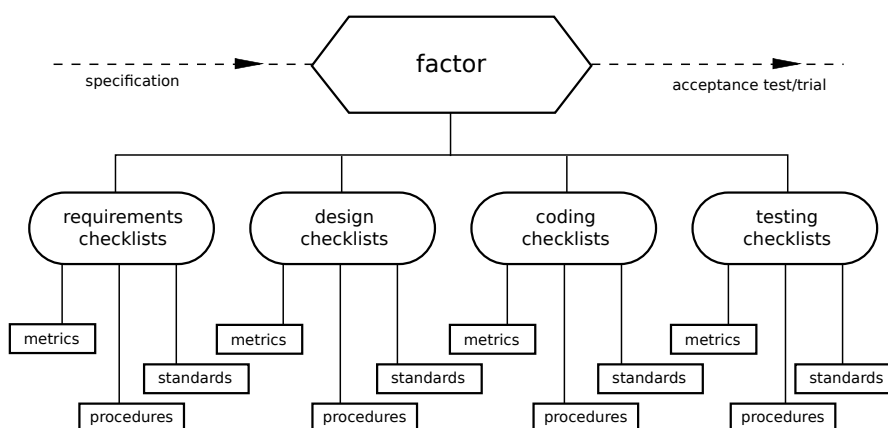


Figura 2.10: Modelo de calidad de COQUAMO.

Los factores de calidad son seleccionados utilizando un procedimiento que consiste en agrupar factores de calidad potenciales en categorías como: factores de calidad específicos de la aplicación, factores de calidad generales que requieren una definición para una aplicación particular, factores de calidad generales definidos de forma independiente de la aplicación y factores de calidad relacionados a la producción de software.

⁷El autor se refiere a estas como “medidas”.

Otro estándar que provee una estructura genérica para la representación de modelos de calidad es el IEEE 1061 [IEEE Std. 1061, 2004] donde se presenta un marco conceptual de métricas de calidad como parte de una metodología (presentada en la sección 2.1.3). El marco propuesto (ver Figura 2.11) se organiza de forma jerárquica en distintos niveles que pueden ser modificados de forma flexible.

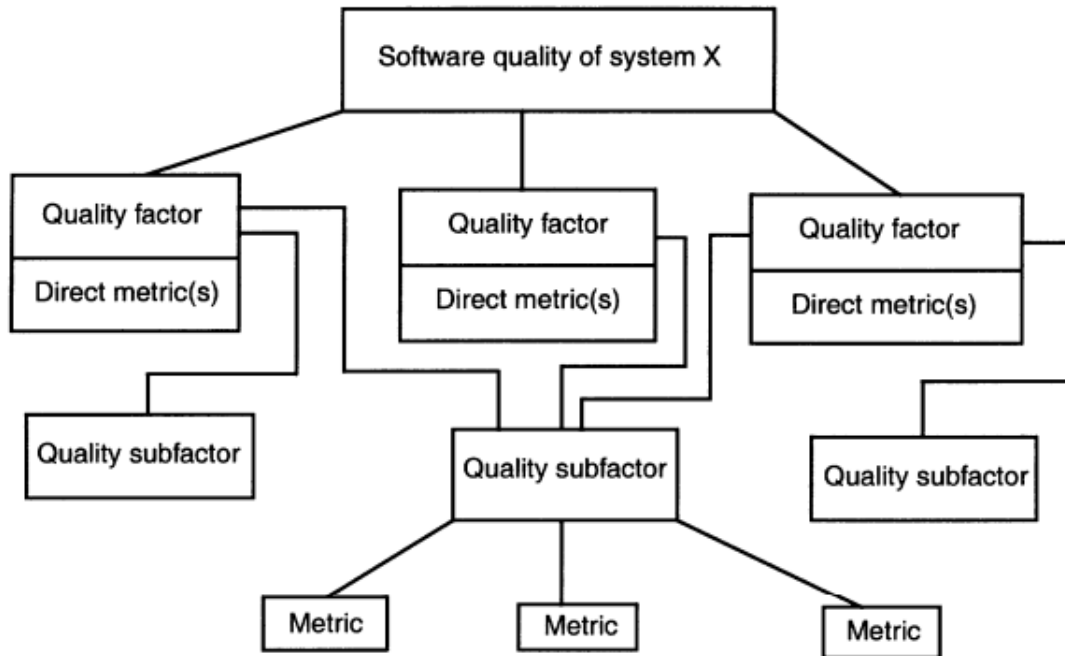


Figura 2.11: Marco conceptual de métricas de calidad del estándar IEEE 1061.

El primer nivel de la jerarquía comienza con el establecimiento de *requerimientos de calidad* asignando *atributos* que describen la calidad del sistema de software a evaluar. A estos atributos se asignan *factores de calidad* que representan vistas de gestión y de usuario. A cada factor de calidad se asignan *métricas directas* que proveen una representación cuantitativa del factor⁸. A cada factor de calidad puede asociarse una o más métricas directas y valores esperados, que permiten determinar si el factor ha sido satisfecho. En el segundo nivel de la jerarquía, los factores de calidad se descomponen en atributos cuantificables del software llamados subfactores de calidad. Estos últimos son independientes del software por lo que pueden corresponder a más de un factor de calidad. En el tercer nivel de la jerarquía los subfactores de calidad se descomponen en métricas de productos y procesos a ser tomadas durante el ciclo de vida de desarrollo.

Otros modelos se enfocan en características de calidad más específicas como es el caso del *Modelo de Eliminación de Defectos Basado en Fases* [Kan, 2002] que resume las relaciones utilizando tres métricas: inserción de defectos, eliminación de defectos y efectividad. El modelo toma como entrada un conjunto de tasas de inserción de errores y de efectividad por fases y modela patrones de eliminación de defectos. Otro ejemplo de estos modelos es el modelo formal de confiabilidad *Rayleigh* [Kan, 2002] utilizado para evaluar la confiabilidad de un producto o estimar el número de defectos latentes cuando se encuentra disponible a los clientes.

Otros modelos se enfocan en evaluar entidades particulares como es el caso de los procesos. El estándar ISO/IEC 15504 (SPICE) [ISO/IEC 15504, 2004] define un marco para la evaluación de procesos de software que cubre las actividades de desarrollo, operación, provisión, mantenimiento y soporte. El estándar define un *modelo de evaluación de procesos* basado en un *modelo de procesos de referencia* – que describe los procesos del ciclo de vida y sus relaciones (definido en el mismo estándar)– que describe

⁸El estándar define métrica directa como “una métrica que no depende de la medida de cualquier otro atributo”.

niveles de capacidad, atributos de proceso y una escala de valoración para cuantificar la capacidad del proceso. Otro de los modelos de evaluación de procesos de amplio uso en la industria es el modelo CMMI (*Capability Maturity Model Integration*) [CMMI Product Team, 2006] que describe un conjunto de áreas de proceso en función de los objetivos y prácticas genéricas y específicas que debieran satisfacer los procesos correspondientes de una organización para alcanzar determinado nivel de madurez. De esta forma, una evaluación de madurez de las capacidades de un proceso según el modelo CMMI se lleva a cabo comprobando la realización efectiva de las prácticas definidas en el modelo.

2.1.3. Enfoques, procesos y métodos

Como en todo área de la Ingeniería de Software, las actividades de medición y evaluación se llevan a cabo siguiendo un conjunto de especificaciones que indican los pasos a realizar, con más o menos detalle, desde las más genéricas que proveen directivas o guías generales hasta las más específicas que proveen detalles de cada tarea a ser realizada (incluyendo recursos, roles, etc.). En la literatura existen diversas publicaciones que describen y/o documentan enfoques, procesos y métodos a ser utilizados en la medición y evaluación de entidades en el marco de la Ingeniería de Software.

2.1.3.1. Enfoques

Los enfoques para medición y evaluación proveen guías que ayudan a seleccionar y priorizar las mediciones que nos proveerán las respuestas que necesitamos. Actualmente se pueden distinguir tres paradigmas en los enfoques utilizados para llevar a cabo la medición y evaluación, según el punto de partida de las actividades. El primero es el “*top-down*” u *orientado a objetivos*, en el que se parte de la necesidad de conocer acerca de un concepto de calidad abstracto y luego se descompone progresivamente en factores y subfactores hasta un nivel cuantificable, es decir, la definición de métricas que luego serán utilizadas para recolectar las medidas de las entidades correspondientes. Este paradigma tiene sus ventajas y desventajas dependiendo del contexto de aplicación. Resulta una buena alternativa ante la ausencia de enfoques o ante enfoques informales o *ad hoc*, si se dispone de recursos limitados y de experiencia en la aplicación de medición de software [Braungarten, 2007]. Su aplicación es simple y permite descubrir rápidamente cuáles son las mediciones que deben ser recolectadas para el objetivo definido. Por otro lado el paradigma también posee desventajas:

- si no se posee una base de mediciones de software que registre el estado actual de la situación, resulta difícil definir de forma consistente los objetivos de alto nivel de antemano, pudiendo omitir aspectos relevantes del fenómeno bajo estudio;
- la cantidad de preguntas formuladas para un objetivo podrían volverse ilimitadas y difíciles de cuantificar mediante métricas de software;
- no se describe cómo interpretar las mediciones obtenidas, es decir, cómo traducir los valores del nivel cuantitativo de vuelta hasta el nivel conceptual.

A pesar de estas desventajas, el paradigma ha demostrado, por su amplio uso en la industria, ser una importante herramienta en diversos contextos organizacionales. Para ser utilizado en entornos ingenieriles, donde se requiere una mayor precisión, se requiere una especificación más detallada de las actividades, roles y artefactos utilizados y producidos, como se verá mas adelante en esta sección. Por su flexibilidad, el paradigma también puede ser complementado por otras técnicas, por ejemplo para la interpretación de los resultados.

El segundo paradigma a considerar es el “*bottom-up*” o “*conjunto fijo de métricas*”. En este paradigma la actividad de medición comienza por el nivel cuantitativo, definiendo y capturando un conjunto de métricas iniciales. Luego mediante la interpretación de las medidas se trata de fundamentar

los resultados buscando posibles problemas e identificando necesidades de mejora para luego definir objetivos y acciones correctivas. Por esto, el paradigma puede ser utilizado para introducir rápidamente la medición de software en una organización. El conjunto inicial de métricas puede establecerse de dos formas: (i) analizando los datos de sistemas legados existentes (o una muestra significativa) para descubrir medidas y sus correspondientes métricas; o (ii) definir un conjunto de métricas significativas enfocándose en los objetos fundamentales del desarrollo de software –los productos de trabajo y los recursos utilizados para crearlos– para lo cual se pueden utilizar guías o recomendaciones definidas para tal propósito, como es el caso de [Carleton *et al.*, 1992]. El principal beneficio de este paradigma es una reducción del esfuerzo y el costo de definir métricas arbitrarias cada vez que comienza un nuevo proyecto. Además promueve la formulación de preguntas técnicas que podrían ser respondidas por los datos en crudo ya disponibles. Por otro lado, un problema del paradigma reside en que, al contar con un conjunto fijo de métricas para todos los proyectos de software, se asume erróneamente que éstos son similares.

El tercer paradigma a considerar es una mezcla de los dos anteriores que, por sus naturalezas, se complementan mutuamente combinando las ventajas de ambos en un paradigma que provee un mecanismo evolutivo, factible y efectivo para llevar a cabo la medición y evaluación en proyectos de software, particularmente en organizaciones con poca o ninguna experiencia en medición de software. Los pasos clave de este paradigma son los siguientes:

- mediante la aplicación del paradigma *top-down* se llega a un conjunto mínimo, lo suficientemente expresivo, de métricas de software,
- mediante la aplicación del paradigma *bottom-up* se integran los datos provenientes de los sistemas legados y se extraen medidas relevantes,
- mediante determinados análisis se busca consolidar los resultados de los pasos anteriores, identificando las medidas resultantes del paradigma *bottom-up* con las métricas definidas en el paradigma *top-down* y examinando estas últimas para determinar si se requieren definir nuevas métricas, no consideradas inicialmente, a partir de las medidas extraídas del paradigma *bottom-up*.

De los tres paradigmas vistos, el más utilizado y documentado en la literatura relacionada es el *top-down*. Los enfoques más representativos de este paradigma son el *Goal-Question-Metric* y el *Balanced Scorecard*.

El enfoque *Goal-Question-Metric* o *GQM* [Basili *et al.*, 1994] propone una estructura jerárquica de tres niveles (ver Figura 2.12) pensada para definir las necesidades de información de la organización y sus proyectos de forma operacional y proveer un marco para interpretar los resultados respecto de los objetivos formulados. El nivel superior o conceptual corresponde a una especificación estructurada de los objetivos que se buscan satisfacer, para los que se especifica el propósito de la medición, el objeto o entidad a ser medido, el factor o característica a ser medida, el punto de vista y el ambiente o contexto donde se lleva a cabo. Posibles entidades a medir incluyen productos, procesos y actividades del ciclo de vida y recursos. Los objetivos son refinados luego a un nivel operacional como una lista de preguntas que buscan caracterizar el objeto medido con respecto a la característica y el punto de vista seleccionados. Finalmente, para cada pregunta se define, a nivel cuantitativo, una o más métricas que permitirán obtener las respuestas a la pregunta correspondiente y, consecuentemente, a los objetivos del nivel conceptual. Las métricas incluyen datos que especifican cómo se obtienen y representan los valores que responden a las preguntas correspondientes. *GQM* prevé que una métrica pueda responder a más de una pregunta para el mismo objetivo, debiendo tener en cuenta que la métrica puede arrojar diferentes valores para diferentes puntos de vista.

Aunque los autores argumentan que el sistema de medición resultante de aplicar el enfoque incluye un conjunto de reglas para la interpretación de los datos de medición, no incluyen detalles o guías de cómo llevarla a cabo.

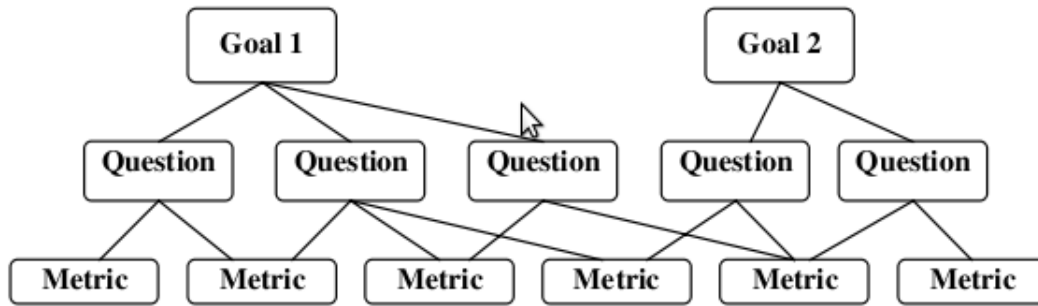


Figura 2.12: Elementos del enfoque GQM.

Otro de los enfoques *top-down* ampliamente reconocidos y utilizados en la industria es el *Balanced Scorecard* (BSC). El BSC es una herramienta de gestión que asiste a una organización en la selección y cuantificación de métricas⁹ que ayuden a conocer de forma cuantitativa el éxito o fracaso de su misión, visión y estrategias [Goethert & Fisher, 2003]. El enfoque propone analizar la estrategia de la organización desde cuatro perspectivas: clientes/usuarios, finanzas, procesos internos y aprendizaje y crecimiento (ver Figura 2.13). Para cada perspectiva se definen objetivos a ser satisfechos para el éxito de la estrategia, métricas que permiten cuantificar la medida en que los objetivos son alcanzados, metas a ser alcanzadas para cada métrica en la perspectiva e iniciativas para lograr dichas metas.

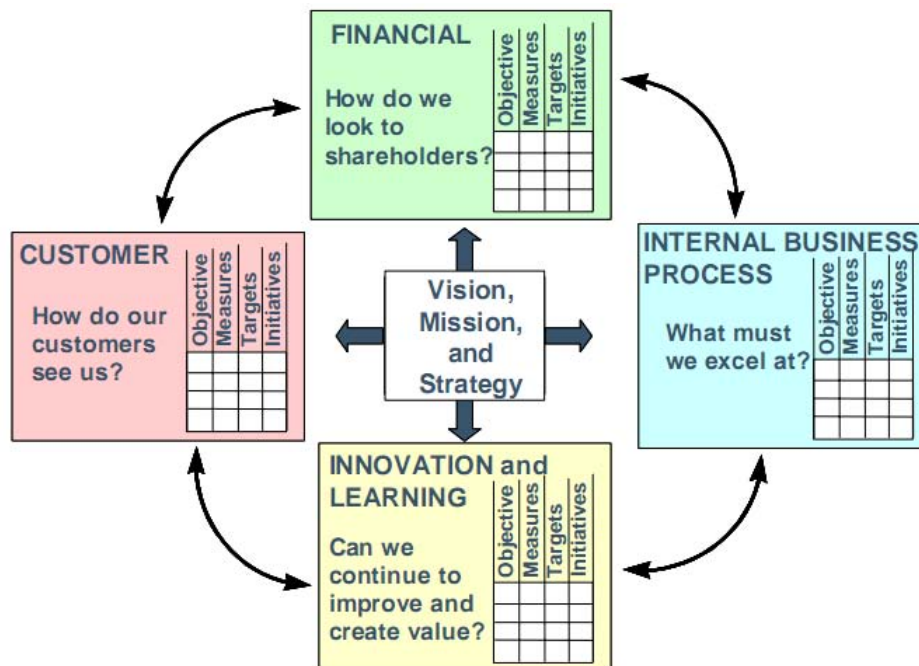


Figura 2.13: Componentes del enfoque BSC.

En esencia, este enfoque es similar al enfoque GQM, sin embargo se observan las siguientes diferencias:

- las perspectivas del BSC hacen referencia a los distintos tipos de entidades u objetos a los cuales apunta la medición, mientras en GQM estos se indican como parte de la especificación de los objetivos;
- en BSC no se provee información acerca de cómo especificar los objetivos;

⁹Los autores se refieren a éstas como medidas.

- en BSC se incluyen los valores esperados (metas) para cada métrica, que proveen un punto de partida para la interpretación de los mismos, y las acciones a llevar a cabo para alcanzar dichos valores, mientras en GQM ambos elementos están ausentes;
- en BSC no existe paralelo al nivel operacional del GQM que asista en el descubrimiento de las métricas relevantes para cada objetivo.

Como se observa en esta comparación, las diferencias de ambos enfoques se complementan, es decir, la desventaja de uno es cubierta por el otro. Sólo en el primer punto, la diferencia resulta una cuestión de forma acerca de cómo se resuelve la especificación de las entidades a medir. Por otro lado, sin embargo, ambos carecen de una descripción de los datos necesarios para especificar claramente una métrica. No obstante las diferencias observadas entre uno y otro enfoque, y lejos de ser aplicados en su forma pura en un contexto ingenieril por su generalidad, ambos plantean un punto de partida para la especificación de procesos y métodos para ser aplicados en proyectos concretos de una organización.

2.1.3.2. Procesos

Un proceso prescribe un conjunto de actividades (y las dependencias entre ellas) a llevar a cabo con un propósito específico, los artefactos que serán utilizados y/o modificados durante su ejecución (por ejemplo, modelos, documentos, código fuente y ejecutables, imágenes, etc.) así como aquellos que se esperan producir y/o modificar como resultado de aplicar el proceso, los recursos que serán necesarios para ejecutar el proceso (por ejemplo herramientas, dispositivos de hardware, equipamientos, ambientes físicos, etc.) y el o los roles (responsabilidades y habilidades) requeridos para llevar a cabo las actividades.

Idealmente, un proceso se define a nivel organizacional y luego es adaptado e implementado para cada proyecto específico de la organización¹⁰. Cuando el proceso se instancia en un proyecto, se deben determinar además los métodos que serán utilizados para implementar cada actividad, los agentes asignados a cada uno de los roles y las restricciones particulares que operen sobre la ejecución de las actividades dependiendo del contexto particular del proyecto.

La especificación de un proceso para medición y evaluación debería incluir todos estos elementos cubriendo actividades como la planificación, gestión, diseño e implementación de tales actividades. En la literatura pueden encontrarse un importante número de estos procesos así como guías para la definición de los mismos.

Uno de los trabajos relevantes del área es el PSM¹¹ (de las siglas en inglés de *Medición de Software Práctica*) [Bailey *et al.*, 2003], una guía para el diseño e implementación de procesos de medición basado en el paradigma *top-down* que incluye además actividades de evaluación de los resultados así como del proceso mismo. La guía incluye un modelo de proceso que define las actividades y tareas necesarias para implementar la medición y evaluación en un proyecto de forma integrada al resto de los procesos técnicos y de gestión de la organización que lo utilizan para su funcionamiento y de donde provienen los requerimientos y necesidades de información de medición. Los resultados de cada actividad (especificaciones de necesidades de información, medidas, procedimientos de recolección y reporte, criterios de evaluación, entre otros) son almacenados en una base de experiencias de medición. Los términos utilizados en la descripción del proceso son definidos en un glosario anexo a la guía.

El proceso incluye cuatro actividades principales:

1. Establecer y sostener un compromiso de medición: se identifican los alcances de la medición y los roles involucrados; se establecen políticas de medición para asegurar la disponibilidad de recursos, la asignación de responsabilidades, entrenamiento, entre otros;

¹⁰Es una condición ideal aunque no estricta ya que una organización en crecimiento puede comenzar a definir e implementar sus procesos a nivel de proyectos para luego elevarlos a la organización cuando hayan sido probados con éxito.

¹¹PSM es patrocinado y utilizado por el Departamento de Defensa de la armada de los EEUU, además de ser utilizado en los programas del gobierno y las industrias, sobre cuya experiencia está basado.

2. Planificar el proceso de medición: se define la necesidad de información y se establecen los procedimientos de recolección, almacenamiento y reporte, así como los criterios de evaluación (tanto para productos como para el proceso de medición mismo);
3. Implementar el proceso de medición: se integran los procedimientos diseñados a los procesos de la organización y se ponen en marcha;
4. Evaluar la medición: se analizan los datos medidos contra los criterios para productos y para el proceso mismo; las lecciones aprendidas son almacenadas en una base de experiencias; se identifican y comunican mejoras potenciales a los productos y al proceso mismo.

PSM también propone un modelo de información (como se verá en la sección 2.1.5) para ser utilizado junto al modelo de proceso los cuales, en conjunto, y según los autores, permiten definir un programa de medición apropiado para cada proyecto particular.

PSM también ha sido utilizado como base para otros desarrollos, como en [Card & Maclver, 2003] donde se describe la aplicación de los conceptos de PSM junto con el enfoque *Balanced Scorecard* para satisfacer las necesidades de información de gestión empresarial. En otro trabajo [Florac *et al.*, 1997], PSM ha sido utilizado como base para definir un proceso de medición para la gestión mejora de procesos.

En la serie de estándares ISO/IEC 14598 [ISO/IEC 14598-1, 1999] se presenta un proceso de evaluación para productos de software, siguiendo el paradigma *top-down* que incluye también las actividades correspondientes a la medición de los mismos. El estándar se compone de seis partes de las cuales la primera contiene los requerimientos generales del proceso para especificar, medir y evaluar la calidad de productos de software durante las etapas del ciclo de vida de software. La segunda parte del estándar define actividades de soporte al proceso tales como planificación y gestión del proceso, entre otras. Las partes 3, 4 y 5 se enfocan en las particularidades del proceso para los casos de desarrollo, adquisición y evaluación independiente o de terceros respectivamente. La parte 6 provee guías para documentar módulos de evaluación, que pueden ser usados para diseñar nuevas evaluaciones. Éste estándar conforma con la terminología y especificaciones del estándar ISO/IEC 9126 visto en la sección 2.1.2. El proceso de evaluación propuesto incluye las siguientes actividades, descritas de forma genérica:

1. Establecer los requerimientos de evaluación: incluye el propósito de la evaluación, los tipos de productos a evaluar (según la fase dentro del ciclo de vida) y un modelo de calidad (en base al estándar ISO/IEC 9126);
2. Especificar la evaluación: seleccionar métricas, establecer niveles de satisfacción para las métricas y establecer los procedimientos para la interpretación de los resultados –integrando los resultados de las métricas en las características del modelo de calidad, como propone el enfoque GQM;
3. Diseñar la evaluación: producir un plan de evaluación que incluye los métodos de evaluación y el cronograma de acciones;
4. Ejecutar la evaluación: tomar las mediciones y compararlas contra los niveles de satisfacción e integrar los resultados para cada característica del modelo de calidad.

Los resultados de cada evaluación son usados para tomar decisiones acerca del siguiente paso en el ciclo de vida de desarrollo de software.

De forma similar al anterior, el estándar ISO/IEC 15504 [ISO/IEC 15504, 2004] se enfoca en la evaluación de procesos, donde se describe un modelo de procesos de referencia y los requerimientos para un modelo de evaluación de procesos. Las diferentes partes del estándar proveen información complementaria sobre las diferentes aplicaciones del modelo de evaluación de procesos.

Por otro lado, en el estándar ISO/IEC 15939 [ISO/IEC 15939, 2001] se define un *Proceso de Medición de Software*, que fue adoptado de la guía PSM y por lo tanto es esencialmente el mismo proceso. Las actividades definidas en este proceso se solapan a las definidas en el estándar ISO/IEC 14598, aunque, a diferencia de este último, se utiliza el término “*medida*” en lugar de “*métrica*”, las cuales se documentan por su nombre, unidad de medición, su definición formal, el método de recolección y su enlace a las necesidades de información.

Relacionados a los anteriores, y más recientes, se encuentran los estándares SQuaRE (de las siglas en inglés de *Requerimientos y Evaluación de Calidad de productos de Software*) [ISO/IEC 25000, 2005] que reemplazan los estándares ISO/IEC 9126 (calidad de productos de software) e ISO/IEC 14598 (evaluación de productos de software) con el propósito de asistir al desarrollo y adquisición de productos de software. SQuaRE cubre la especificación de requerimientos de software y la evaluación de calidad de software, soportados por un proceso de medición de calidad de software. En este último proceso, SQuaRE adopta el estándar ISO/IEC 15939 en la forma de un *Modelo de Referencia de Medición de Calidad* de productos de software. El modelo provisto por SQuaRE permite alinear las definiciones de calidad de los clientes con los atributos del proceso de desarrollo. Además provee métricas¹² de atributos de calidad de productos de software que pueden ser usadas por desarrolladores, compradores y evaluadores externos. SQuaRE está compuesto de cinco divisiones o subconjuntos de estándares, de los cuales, los siguientes están directamente relacionados con los procesos de medición y evaluación:

- ISO/IEC 2502n - División de Medición de Calidad. Los estándares de esta división incluyen un modelo de referencia de medición de calidad de productos de software [ISO/IEC 25020, 2007], definiciones matemáticas de métricas de calidad y guías prácticas para su aplicación. Provee ejemplos de calidad de software interna, externa y de calidad en uso y se definen y presentan los elementos de las métricas correspondientes.
- ISO/IEC 2504n - División de Evaluación de Calidad. Los estándares de esta división proveen requerimientos, recomendaciones y guías para la evaluación de productos de software, para ser utilizados por evaluadores, compradores o desarrolladores. También se incluye el soporte para documentar módulos de evaluación.

Al momento de escritura de este capítulo, sólo algunos de los estándares que componen SQuaRE han sido publicados.

Otro modelo utilizado en la industria es el *CMMI for Development* [CMMI Product Team, 2006], introducido en la sección 1.1.1. Este modelo no define procesos en sí sino que describe un conjunto de áreas de proceso en función de los objetivos y prácticas genéricas y específicas que debieran satisfacer los procesos correspondientes de una organización para alcanzar determinado nivel de madurez. El área de proceso de CMMI relevante a este trabajo es el de *Medición y Análisis* destinada a desarrollar y mantener una capacidad de medición para soportar las necesidades de información de gestión. Para ésta área el modelo propone satisfacer dos objetivos genéricos, cada uno con un conjunto de prácticas específicas:

- Alinear las actividades de medición y análisis con objetivos y necesidades de información:
 - Establecer y mantener objetivos de medición derivados de necesidades de información;
 - Especificar medidas cuantificables para satisfacer los objetivos de medición;
 - Especificar procedimientos de recolección y almacenamiento de los datos de las mediciones;
 - Especificar procedimientos de análisis y reporte de los resultados.

¹²Aunque SQuaRE reemplaza los estándares ISO/IEC 9126 y 14598 adopta el término *medida* utilizado en el ISO/IEC 15939.

- Proveer los resultados de mediciones que respondan a las necesidades especificadas:
 - Recolectar los datos de medición;
 - Analizar e interpretar los datos de medición;
 - Gestionar y almacenar los datos de medición, las especificaciones de medición y los resultados del análisis;
 - Comunicar los resultados de las mediciones y análisis a todos los agentes interesados.

Como se observa, las actividades definidas para el área de *Medición y Análisis* de CMMI siguen el paradigma *top-down*, y ya que no establecen requerimientos específicos en las dependencias de las actividades, los objetivos y prácticas propuestas pueden ser satisfechas por cualquiera de los procesos vistos.

Otros autores han utilizado el enfoque GQM como base para la definición de procesos tal como en [Gresse *et al.*, 1995] y [McAndrews, 1993] donde se describen modelos de proceso para llevar a cabo la medición y análisis en proyectos de software de una organización. En ambos casos, los autores describen el conjunto de actividades que conforman el proceso que, básicamente, consisten en aquellas definidas por el enfoque GQM. No obstante, ambos trabajos presentan características distintivas. Puntualmente, en [Gresse *et al.*, 1995] se destaca una etapa inicial de *pre-estudio* donde se caracteriza a la organización y sus proyectos para luego identificar los objetivos de mejora. También se destaca el uso de un almacenamiento de datos de medición cuyos procedimientos para su uso deben ser especificados en un plan GQM, así como el almacenamiento de las experiencias y resultados documentados junto con la descripción del contexto en el cual el modelo de medición es válido y/o puede ser aplicado. Por otro lado, en [McAndrews, 1993] se destaca, durante la planificación, la definición de las técnicas de análisis a utilizar. También durante la implementación del proceso se destaca la revisión de los procedimientos utilizados para determinar su adecuación a las particularidades del proyecto. Finalmente, se incluye una actividad de *evolución del proceso* donde los reportes de medición son revisados evaluando el progreso logrado e identificando y proponiendo ajustes de mejora a los procedimientos.

Otros autores han definido procesos que proponen mejoras al enfoque GQM original como es el caso de [Park *et al.*, 1996] en el que se destaca el uso de “indicadores”, incorporados en el modelo de datos GQM entre el nivel operacional (preguntas) y cuantitativo (métricas) con el objetivo de proveer una vista gráfica del valor del atributo que ayudaría a responder la pregunta correspondiente. Los autores distinguen esta incorporación del enfoque original con el nombre de GQ(I)M. El proceso consta de un número de pasos que describen de forma detallada las actividades a realizar y las plantillas y formatos utilizados para registrar los productos intermedios del proceso de medición y análisis siguiendo el enfoque mencionado.

En [Briand *et al.*, 2002] se define un proceso para definir métricas¹³ de atributos de productos teóricamente válidas siguiendo el enfoque GQM. El proceso se resume en los siguientes pasos de alto nivel:

1. Los objetivos de la organización son refinados en objetivos de medición basándose en conocimiento del entorno (equipos de desarrollo y experiencias), identificando procesos y productos a medir. De estos objetivos, se establecen hipótesis empíricas sobre aspectos de calidad, usualmente externa, tales como confiabilidad, mantenibilidad, esfuerzo, etc.;
2. Se definen las medidas para los atributos independientes y dependientes, formalizando las entidades de donde se extraen, para el contexto particular de la aplicación;

¹³Los autores utilizan el término “medida”.

3. Las hipótesis empíricas son verificadas y refinadas estableciendo relaciones funcionales entre las medidas obtenidas. Con las nuevas hipótesis se construyen modelos de predicción para verificar la plausibilidad de nuevas hipótesis empíricas.

Otros procesos definen actividades de medición y evaluación destinadas a la mejora de procesos. Uno de ellos es el definido en el proyecto AMI [Debou *et al.*, 1994] (de las siglas en inglés *Aplicación de Métricas en la Industria*) que propone 4 actividades:

1. Evaluación del contexto del proyecto y definición de los objetivos primarios de medición;
2. Análisis de los objetivos primarios y derivación de los subobjetivos hasta el nivel cuantitativo, siguiendo el modelo de preguntas y métricas del enfoque GQM;
3. Recolección y validación de los datos siguiendo las especificaciones de las métricas definidas en la actividad anterior;
4. Los datos de medición son comparados con los objetivos y preguntas del plan de medición para determinar si los objetivos del proyecto han sido alcanzados; en caso negativo se implementan las acciones necesarias, de lo contrario se reevalúan los objetivos primarios para continuar con la mejora.

En [Braungarten, 2007] se presenta un modelo de proceso de medición, SMPI (de las siglas en inglés de *Mejora del Proceso de Medición de Software*), que integra los tres paradigmas de medición (vistos anteriormente) para permitir la mejora del proceso desde un estado inicial en el que la organización no posee una base de medidas de su actividad. El modelo de proceso se sitúa en el contexto del área de proceso de *Medición y Análisis* de CMMI (v1.1), utilizada como base para incorporar la aplicación secuencial de los paradigmas '*bottom-up*', '*mixed*', y '*top-down*' durante la implementación del proceso de medición. El modelo propuesto provee tres submodelos para cada paradigma definiendo las dependencias correspondientes a las transiciones de paradigma. El modelo permite iniciar el proceso de medición utilizando cualquiera de los tres paradigmas, dependiendo de las condiciones iniciales de la organización.

2.1.3.3. Métodos

Como se mencionó anteriormente, cuando se ejecuta un proyecto, se deben decidir los métodos que serán utilizados para realizar cada actividad enunciada por el o los procesos implementados. Los métodos establecen cómo realizar una determinada actividad o tarea prescribiendo un conjunto de pasos, modelos, herramientas, entre otros, como guía práctica para llevar a cabo la actividad para la cual fue diseñado. De forma análoga, una metodología establece un familia de métodos para realizar un conjunto de actividades relacionadas. Así, existen en la literatura diversos métodos y metodologías para las actividades involucradas en la especificación de requerimientos de calidad, el diseño e implementación de la medición y de la evaluación, y para el análisis y reporte de resultados.

En el estándar IEEE 1061 [IEEE Std. 1061, 2004]¹⁴ se documenta una *Metodología para Métricas de Calidad de Software* que describe los pasos para establecer requerimientos de calidad e identificar, implementar, analizar y validar métricas de calidad de software de productos y procesos. Esta metodología puede aplicarse a todas las fases de cualquier ciclo de vida. La metodología comprende cinco pasos que deben ser aplicados de forma iterativa y siguen básicamente las etapas del paradigma *top-down*: establecer requerimientos, identificar, definir e implementar las métricas y analizar los resultados. En la última etapa se validan las métricas para identificar aquellas que pudieran predecir valores de factores de calidad específicos para procesos y productos y ser utilizadas en etapas tempranas del ciclo de vida

¹⁴La primer versión fue creada en 1992, luego revisada en 1998 y reafirmada en 2004.

cuando los valores reales de los factores de calidad no están disponibles. La validación debe ser realizada en ambientes de desarrollo estables y deben ser aplicadas en proyectos similares (esta condición será analizada con mayor profundidad en la sección 2.1.6).

En [van Solingen & Berghout, 1999] se presenta el Método GQM, una guía práctica para asistir a la implementación de programas de medición siguiendo el enfoque homónimo. El método presentado contiene cuatro fases: planificación, definición, recolección de datos e interpretación, cada una detallando el conjunto de pasos a seguir.

En [Olsina, 1999; Olsina & Rossi, 2002] se presenta WebQEM, de las siglas en inglés de Metodología de Evaluación de Calidad Web, que describe un conjunto de fases, actividades y métodos para llevar a cabo la medición y evaluación de requerimientos no funcionales en aplicaciones web. La metodología cuenta con soporte tecnológico y ha sido aplicado en la evaluación de calidad de sitios web. El proceso que subyace a la metodología sigue el paradigma *top-down* y está basado en el modelo de proceso para evaluadores del ISO [ISO/IEC 14598-1, 1999] (visto anteriormente en la sección 2.1.3.2). El método consta básicamente de cuatro fases, a saber: *definición y especificación de requerimientos de calidad*, *evaluación elemental*, *evaluación parcial y global* y *conclusión y recomendaciones*. A partir de este trabajo surgen nuevas líneas de investigación con el objetivo de formalizar una conceptualización de los términos utilizados y del proceso subyacente a la metodología así como trabajos relacionados a su aplicación y a la definición de memorias organizacionales basadas en casos, como se verá en secciones subsiguientes.

En [Rodríguez *et al.*, 2010] se presenta una metodología para la evaluación continua de calidad de artefactos de software. La metodología (CQA-Meth) establece un marco para el establecimiento de los procesos necesarios para llevar a cabo la evaluación de cualquier artefacto de software, así como mantener la comunicación entre el cliente y el equipo de evaluación. Junto a la metodología, se propone un conjunto de herramientas que la soportan: una herramienta común que soporta la metodología (CQA-Tool) más un número de herramientas específicas para cada artefacto en particular. La herramienta principal permite construir un catálogo de técnicas de evaluación que integra las herramientas disponibles para cada artefacto de software (ej. métricas, listas de comprobación, convenciones de modelado, guías, etc.). La metodología se compone de 3 procesos principales, uno de los cuales consiste en la evaluación misma, involucrando las fases de planificación, especificación, ejecución y conclusión. Los dos procesos restantes se enfocan en la gestión de la evaluación (documentación, control y evaluación del proceso) y la gestión de la infraestructura (especificación, mantenimiento, adaptación y transferencia de la infraestructura disponible para el proceso de evaluación).

Otros métodos se enfocan en aspectos particulares de la medición y evaluación en proyectos de software. Tal es el caso de [Goethert & Fisher, 2003] donde se describe la aplicación sinérgica e iterativa del Balanced Scorecard y de GQM para desarrollar métricas¹⁵ e indicadores asociados para medir el desempeño de una organización. Según la propuesta, los objetivos y subobjetivos estratégicos de la organización son derivados y priorizados utilizando GQ(I)M; luego los subobjetivos son mapeados al BSC y refinados iterativamente, definiendo criterios de éxito para cada uno; luego, utilizando el método GQ(I)M, se derivan métricas e indicadores para cada subobjetivo y cada cuadrante/dimensión del BSC.

En [Florac, 1992] se describen mecanismos para describir y especificar métricas de problemas y defectos de software, utilizadas para entender y predecir la calidad de productos y la eficacia de los procesos de software. El marco propuesto por los autores soporta el descubrimiento, reporte y medición de problemas y defectos de software al proveer:

- diferentes actividades que permiten descubrir problemas en los diferentes artefactos de software presentes durante el ciclo de vida de software;

¹⁵Los autores utilizan el término “medida”.

- los diferentes atributos medibles que pueden utilizarse para caracterizar problemas y defectos del software;
- los mecanismos a utilizar para comunicar la definición y especificación, de forma clara y repetible, de mediciones de problemas y defectos de software. Estos mecanismos están basados en listas de comprobación especialmente estructuradas para tal fin;
- una guía que ilustra, mediante formularios apropiados, cómo obtener y especificar cada uno de los atributos identificados utilizando los formularios diseñados, así como para utilizar los atributos definidos en la medición integral de problemas y defectos en proyectos de software.

En [Schubert & Dettling, 2002] se presenta un método y una herramienta para la evaluación de aplicaciones de comercio electrónico. La evaluación se basa en una grilla que incluye un conjunto de criterios de calidad específicos del dominio y se enfoca en la perspectiva del usuario. Para cubrir las diferencias de los diferentes sectores, los criterios son ponderados respecto de su relevancia en los mismos. Los criterios de calidad son asignados a una categoría (facilidad de uso, utilidad, confianza) y a una de las cuatro fases de transacción del comercio electrónico –información, acuerdo, liquidación y post-venta– o a la categoría “todas las fases”. Durante la recolección de datos, cada criterio es valuado en una escala de 4 posibles valores expresando una valuación positiva o negativa. Los datos son luego evaluados efectuando la comparación contra tres perfiles diferentes: el promedio del sector, la mejor práctica o alguna otra evaluación puntual, para lo cual los autores proveen los cálculos que se llevan a cabo. Finalmente se genera un reporte conteniendo análisis y representaciones gráficas de los resultados individuales de los criterios y de las comparaciones con los diferentes perfiles.

En [Stefani *et al.*, 2003] se propone un método basado en redes bayesianas para modelar, evaluar y definir requerimientos de calidad –durante el diseño– de sistemas de comercio electrónico. El modelo provee estimaciones basadas en experiencias pasadas provenientes de aplicaciones de comercio electrónico, demandas de usuarios finales y características de calidad. Las estimaciones consisten en probabilidades discretas que cambian a medida que se incorpora nueva evidencia al modelo. El modelo utiliza nodos interconectados para representar factores de calidad, características y subcaracterísticas (basadas en el estándar ISO 9126) de los sistemas de comercio electrónico. Cada nodo es caracterizado por un valor de evidencia (tomado de un conjunto de estados posibles –bueno, promedio y pobre) que indica la presencia de una determinada característica o subcaracterística en el sistema. El modelo es utilizado ingresando los valores de nueva evidencia en algunos nodos que, junto a la evidencia de experiencias previas, afecta las probabilidades de los otros nodos, actualizando la estimación final. El modelo puede ser utilizado de forma directa para evaluar la calidad global o parcial (acerca de alguno de los factores de calidad) de un sistema ingresando en cada nodo del modelo la evidencia disponible (mediciones) y arrojando una estimación con los valores correspondientes de probabilidad, o de forma inversa para proveer estimaciones acerca de los nodos intermedios cuando se conoce el valor final de calidad.

En [Dujmovic, 1996, 2007] se presenta un método de decisión cuantitativo general –y su modelo asociado– para la evaluación, comparación y selección de sistemas de software y hardware complejos (llamado *Logic Scoring of Preference* o LSP). El método permite representar todas las propiedades deseables o requerimientos en la entidad a evaluar mediante funciones complejas de múltiples criterios usando una lógica de preferencia continua. Para cualquier tipo de entidad la aplicación del método supone la existencia de los objetivos de evaluación a partir de los cuales se especifican un conjunto de variables de desempeño o requerimientos para las propiedades relevantes del sistema, que pueden ser agrupadas en más de un nivel de características más abstractas. Luego, para cada variable de desempeño se define un rango de valores aceptables y una función –*criterio elemental*, que define cómo convertir valores de la variable de desempeño a valores normalizados de preferencia elemental que indican el grado o porcentaje de satisfacción del requerimiento correspondiente. Finalmente se

especifica una estructura de agregación de preferencias donde los valores de preferencia elemental son agregados de forma incremental para generar un valor de preferencia parcial y finalmente global, que representa el grado de satisfacción de los requerimientos como un todo. La función de agregación se organiza combinando una suma de potencias ponderadas, utilizando una potencia apropiada para lograr las propiedades lógicas deseadas en un rango que va desde la disyunción hasta la conjunción.

2.1.3.4. Otros enfoques, prácticas y métodos

Por otro lado, existe en la literatura un importante número de publicaciones que abarcan desde aspectos fundamentales sobre la medición y evaluación de software hasta procesos, métodos, técnicas, enfoques, métricas y prácticas recomendadas para la implementación en programas de medición. Tal es el caso de [Fenton & Pfleeger, 1997] donde se tratan los fundamentos de la medición, experimentación, recolección y análisis de datos, se presenta información detallada de métricas para los aspectos relevantes de la Ingeniería de Software y se trata la perspectiva de la gestión de programas de medición, incluyendo casos de éxito y guías para evaluar la efectividad de técnicas y herramientas. También en [Zuse, 1998] se presentan aspectos generales y propiedades deseables de la medición de software así como mecanismos de validación y predicción de mediciones, entre otras aplicaciones.

En [Kan, 2002] se describe qué es calidad de software y se presentan fundamentos de la teoría de medición. Además los autores presentan aplicaciones de herramientas de calidad al desarrollo de software, modelos y métricas de calidad en diferentes aspectos del desarrollo de software, modelos de gestión de calidad, integración de evaluaciones de calidad en los procesos y proyectos, prácticas recomendadas y no recomendadas, entre otros.

En [Ebert & Dumke, 2007] se describen procesos y prácticas de medición de software describiendo los componentes de un programa de medición y la infraestructura asociada, incluyendo la gestión y control de proyectos, control y aseguramiento de calidad y la mejora de calidad.

2.1.4. Elementos de la medición y evaluación

Como se vio en la Sección 2.1.3, para llevar a cabo la medición y evaluación de calidad de un producto de software es necesario especificar un conjunto de elementos que describen y definen las actividades a realizar. Estos elementos de información son documentados como parte de la planificación formal del proceso de medición y evaluación, y son utilizados a lo largo de todo el ciclo de vida del producto para asegurar la satisfacción de los requerimientos de calidad. La descripción y especificación estructurada de estos elementos puede ser utilizada para desarrollar formularios o herramientas de software para dar soporte a la planificación y recolección de datos. A continuación se describen, a modo de resumen, los elementos de información introducidos en la subsección anterior (los conceptos enumerados y descriptos establecen el vocabulario que se utiliza en el resto de este trabajo):

- Objetivo de medición o necesidad de información que debe identificar de forma clara el área o problemática a mejorar y describir el *porqué* se mide; incluye el propósito por el cual se lleva a cabo la evaluación –por ejemplo comprender, mejorar, evaluar, predecir, etc.–, el foco, perspectiva o aspecto de calidad a evaluar –por ejemplo, calidad, calidad en uso, eficiencia, confiabilidad, etc.–, el objeto a evaluar –por ejemplo, un producto, recurso, proceso, etc.–, el punto de vista desde el cual se considera el foco de la evaluación –por ejemplo, el usuario final, el desarrollador, el líder de proyecto, etc.;
- Entidades a medir, asociadas y/o que influyen al objeto a evaluar;
- Modelo de calidad que especifica los requerimientos de calidad del objeto a evaluar, descrito en función de características, subcaracterísticas y atributos cuantificables de las entidades correspondientes;

- Mecanismos de medición y evaluación, que incluyen:
 - Métricas que especifican cómo cuantificar cada uno de los atributos de las entidades a medir. Las métricas incluyen el método o procedimiento de recolección de datos (medición directa o indirecta) pudiendo indicar el uso de determinadas herramientas, la escala y unidad con los que se representarán los valores obtenidos, el momento en el tiempo y frecuencia en que deberán ser realizadas las mediciones, el o los roles del personal responsable de llevar a cabo la recolección;
 - Indicadores que especifican cómo interpretar los valores medidos de los atributos para las características de alto nivel del modelo de calidad y finalmente para el objetivo de evaluación. Los valores medidos son interpretados a una nueva escala mediante modelos (pudiendo ser representados gráficamente) que describen distintos niveles de aceptabilidad para los valores esperados para cada atributo/pregunta. Las características de alto nivel son interpretadas de forma similar mediante modelos de agregación de los valores interpretados para las subcaracterísticas. Adicionalmente, pueden especificarse herramientas y técnicas de análisis para realizar la interpretación;
- Descripción o caracterización del contexto o entorno de la entidad relevante al objetivo de medición, es decir aquellos factores de la organización, proyecto, procesos, recursos y otras entidades relacionadas que influyan en el objetivo de medición. Ejemplos de estos factores son aquellos del entorno en el cual se lleva a cabo la medición, de los métodos y técnicas utilizadas en el ciclo de vida del producto, las características del equipo de desarrollo, factores que pudieran influir en el resultado de las métricas. Todos estos factores del contexto debieran también ser definidos como métricas a ser medidas como parte del proceso de medición y evaluación de la entidad bajo análisis [van Solingen & Berghout, 1999];
- Resultados de mediciones y evaluaciones debidamente almacenadas –con referencias a las especificaciones que dieron lugar a los mismos– para su fácil acceso, a fines de ser utilizado en la toma de decisiones certera y coherente.

Los elementos de información mencionados anteriormente debieran ser cubiertos por modelos de información que especifiquen los detalles necesarios para documentar el diseño e implementación de las actividades de medición y evaluación. A continuación se describen y comentan los modelos de información de medición y evaluación encontrados en la literatura.

2.1.5. Modelos de información para medición y evaluación

En el Capítulo 1 se mencionó la ventaja de contar con un modelo de información que defina cómo especificar la información necesaria para llevar adelante actividades de medición y evaluación. En la literatura existe un número de propuestas de modelos de información de medición y evaluación, aunque en menor medida que la cantidad de propuestas sobre procesos para tal fin, y en muchos casos careciendo de los elementos esenciales para lograr una especificación clara y coherente de las actividades de medición y evaluación.

En [Card, 2000] se presenta un marco de medición y evaluación que incluye entre sus componentes un modelo de información basado en la estructura general propuesta por el enfoque GQM (tratado en la sección 2.1.3) y en la terminología provista por el vocabulario internacional de metrología [ISO/IEC 99:1993, 1993] –reemplazando el término “cantidad” por el de “medida”. El modelo propuesto (presentado en la Figura 2.14) incluye los conceptos y relaciones de los principales elementos de medición y evaluación pero no se describen los mismos de forma detallada y estructurada. Este modelo fue adaptado e incluido en el estándar ISO/IEC 15939 [ISO/IEC 15939, 2001] y adoptado por PSM (*Practical*

Software Measurement) [Card & Maclver, 2003]. En ninguno de éstos últimos se proveen definiciones o especificaciones adicionales al modelo de información propuesto que describan de forma clara y estructurada sus componentes. En el modelo propuesto los atributos de las entidades de software (procesos y productos que son objeto de medición) se relacionan con las necesidades de información del usuario mediante tres niveles de medidas o cantidades –pudiendo agregar elementos en cada uno de los niveles (aunque los autores no aclaran cómo):

- Medidas base: cuantifican un atributo simple aplicando un método de medición;
- Medidas derivadas: combinan dos o más medidas base usando una función matemática;
- Indicadores: pueden ser una medida base o derivada o una combinación de tales medidas asociadas a criterios de decisión por medio de un modelo matemático o heurístico;

Finalmente, según el modelo, un producto de información consiste de uno o más indicadores con interpretaciones correspondientes formando la base para la toma de decisiones.

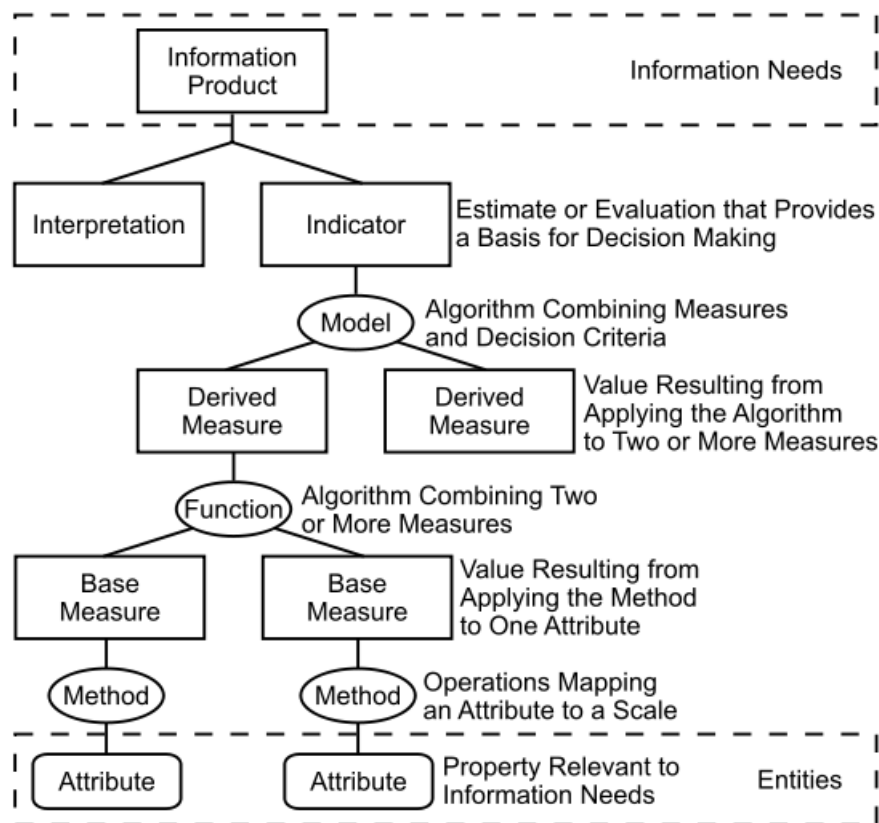


Figura 2.14: Modelo de Información de PSM.

En [Kitchenham *et al.*, 2001] se presenta un modelo conceptual que captura los conceptos y relaciones de los datos utilizados en la medición y evaluación de software mediante un diagrama Entidad-Relación. Así, es posible generar bases de datos que soporten directamente el modelo de datos propuesto y almacenar de forma estructurada los valores de medición junto a los metadatos (definiciones estructuradas de los conceptos utilizados) asociados, durante la implementación de programas de medición de software. Los autores indican que el uso de tal estrategia permite asegurar que tales datos sean confiables, repetibles y comparables haciendo posible:

- efectuar la validación de los datos, durante la entrada de los mismos,

- asegurar la consistencia de las conclusiones, durante el análisis de datos,
- guiar las actualizaciones apropiadas al almacén de datos a partir de los cambios efectuados a los metadatos.

El modelo propuesto (ver Figura 2.15) define un conjunto de conceptos en términos de tres dominios diferentes; un dominio genérico que incluye atributos, unidades y escalas asociadas definidos de forma independiente; un dominio del modelo de desarrollo que traduce atributos y unidades a medidas que son enlazadas a tipos de entidades de software; y un dominio de proyecto que incluye entidades reales, instancias de los tipos de entidades correspondientes, enlazadas a los valores medidos generados a partir de las medidas definidas. Los elementos restantes del modelo permiten describir los elementos mencionados. Los primeros dos dominios definen los metadatos necesarios para especificar (de forma completa y precisa, según afirman los autores) los datos de medición. El dominio de proyecto trata con la recolección de datos de medición (y su automatización) permitiendo que sean almacenados en un formato derivado de los metadatos.

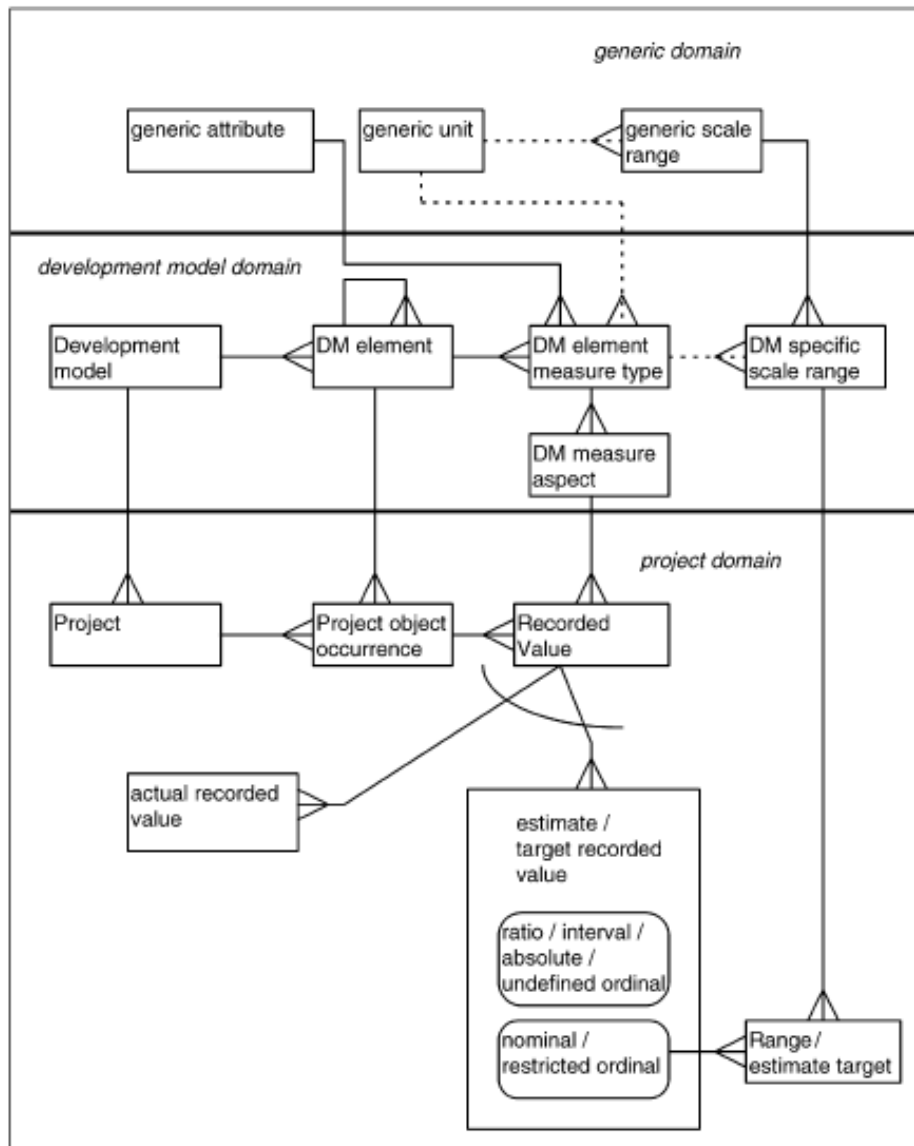


Figura 2.15: Modelo de información de medición presentado en [Kitchenham *et al.*, 2001].

Briand *et al.* proponen una formalización de los conceptos involucrados en su proceso GQM/MEDEA [Briand *et al.*, 2002] (introducido en la Subsección 2.1.3) utilizando un diagrama de clases en UML

(Unified Modeling Language) [OMG-UML, 2012] como punto de partida para el diseño orientado a objetos de una herramienta para soportar la metodología y el reuso de la información del programa de medición (ver Figura 2.16). El modelo incluye los elementos de medición mencionados en la subsección anterior, e incluye otros adicionales como los relacionados al programa de medición y sus recursos. Sin embargo, aunque los elementos son descriptos y definidos a lo largo del texto, no se provee una especificación estructurada y concisa de estos conceptos. Adicionalmente no existe una clara distinción entre la definición de los procedimientos de medición y los valores medidos correspondientes.

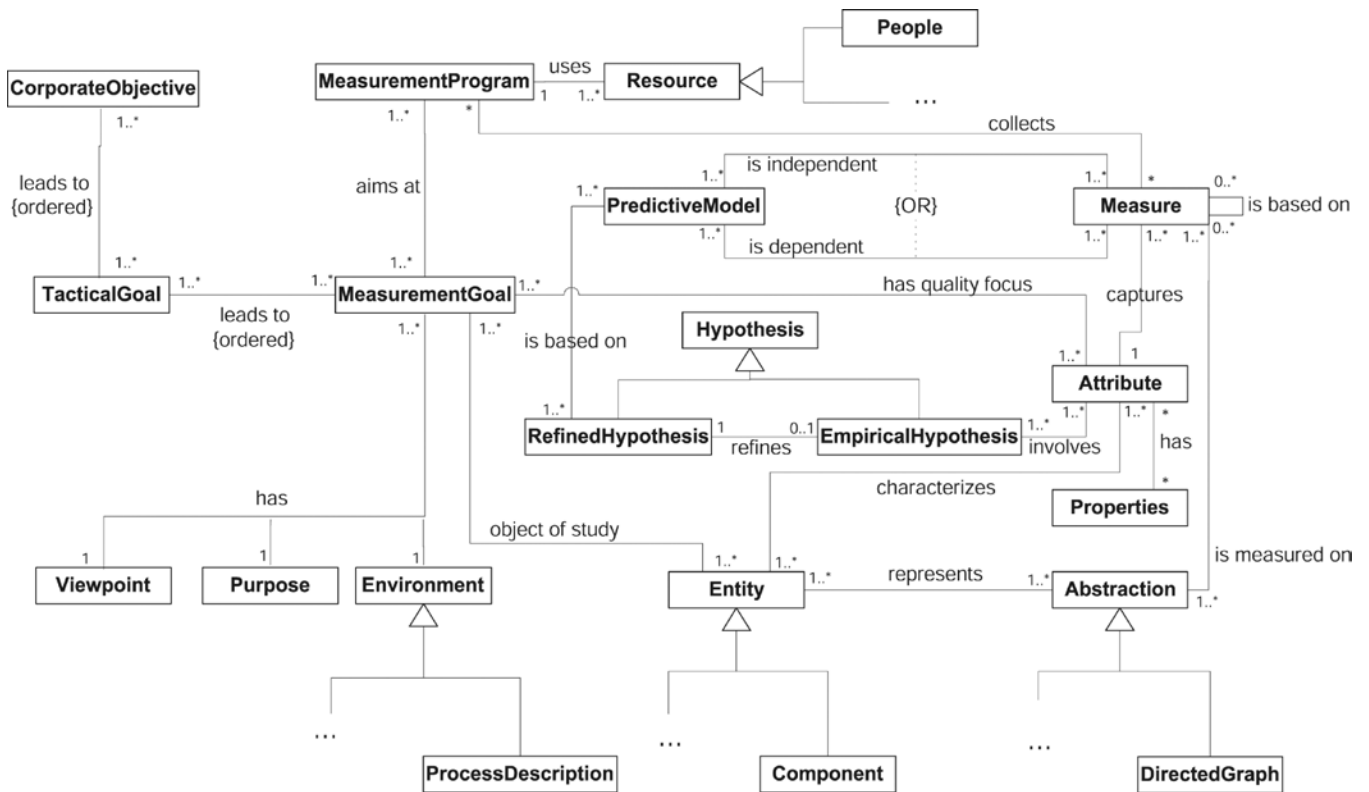


Figura 2.16: Modelo conceptual de medición de Briand *et al.*

En [Lawler & Kitchenham, 2003] se describe una tecnología para el modelado de medición, basada en trabajos previos de Kitchenham *et al.* [Kitchenham *et al.*, 2001], que permite automatizar completamente un programa de medición integrándolo a herramientas de soporte al desarrollo de software así como realizar una comparación confiable entre proyectos, departamentos y compañías. Esta tecnología incluye:

- Una definición rigurosa de las mediciones atómicas a ser recolectadas y de las métricas (definidas como funciones de medidas atómicas) a ser calculadas y reportadas;
- Una definición de los enlaces entre las medidas y las herramientas de soporte al proceso de desarrollo que permite extraer o generar las medidas atómicas;
- Esquemas de medición y modelos de objetos de medición son usados para definir la estructura de la base de datos. El diseño de la base de datos almacena cada esquema de medición y sus instancias.

La propuesta de modelado incluye los siguientes elementos de información:

- *Fuente de medición*: representa un grupo de entidades relacionadas de alguna forma;
- *Medición atómica*: aquella que puede ser obtenida directamente de una entidad, y para la cual los autores presentan una especificación estructurada;
- *Grupo de medición*: agrupa un conjunto de mediciones atómicas y se asocia a una entidad individual dentro de una fuente de medición;
- *Esquema de medición*: agrupa un conjunto de grupos de medición (y las relaciones entre ellos) para una fuente de medición dada;
- *Modelo de objetos de medición*: combina un conjunto de esquemas de medición incluyendo las relaciones entre los grupos de medición de cada esquema;
- *Métrica*: función o expresión matemática que combina una o más medidas atómicas u otras métricas. Una métrica se asocia a un grupo de medición dentro de un modelo de objetos de medición particular. Estas son definidas por los autores de forma estructurada aunque de forma parcial;
- *Estructura de descomposición de medición*: es la instanciación de un modelo de objetos de medición.

No obstante los beneficios enunciados por los autores, aunque todos estos elementos son descriptos, no se presenta una definición estructurada de los mismos y solo se utiliza una notación gráfica propia para describir las relaciones entre dichos elementos.

Otros autores han tratado de especificar los conceptos relacionados a la medición y evaluación de software en ontologías que definan de forma clara los términos, propiedades y relaciones correspondientes a los elementos de información enumerados en la sección 2.1.4. Tal es el caso de [Martín & Olsina, 2003; Olsina & Martín, 2004] donde se presenta una ontología de métricas e indicadores de software con el objetivo de dar soporte a diferentes procesos, métodos y herramientas, particularmente para un sistema de catalogación web, también presentado por los autores. La propuesta incluye un diagrama donde se especifican de forma estructurada todos los conceptos de la ontología, las propiedades que los caracterizan y las relaciones entre ellos (ver Figura 2.17) así como las definiciones de cada uno de ellos. La ontología propuesta continúa una línea de investigación previa donde se desarrolla una metodología para la evaluación de calidad de sitios web [Olsina, 1999; Olsina & Rossi, 2002]. La ontología está basada en los conceptos de los estándares ISO relacionados a modelos calidad de software [ISO/IEC 9126-1, 2001] y procesos de medición y evaluación [ISO/IEC 15939, 2001; ISO/IEC 14598-1, 1999], motivada por la observación de una falta de consenso entre los mismos términos de estos documentos, o de términos ausentes. La ontología también se apoya en conceptos de otras fuentes tales como los trabajos de Zuse [Zuse, 1998], Kitchenham et al. [Kitchenham *et al.*, 2001] y Briand et al. [Briand *et al.*, 2002]. Los autores enfatizan con esta propuesta la importancia de asegurar la consistencia y comparabilidad de los metadatos y datos de medición de software.

La ontología cubre la definición de los elementos de información relacionados a los siguientes aspectos:

- Especificación de requerimientos no funcionales: involucra la definición de una necesidad de información para una entidad particular y un foco de calidad en la forma de un concepto calculable; éstos son representados mediante un modelo de conceptos que relaciona otros conceptos calculables y atributos cuantificables de la entidad;
- Diseño e implementación de la medición: involucra la definición de métricas (directas e indirectas), especificadas mediante una escala categórica o numérica, incluyendo la unidad de los valores

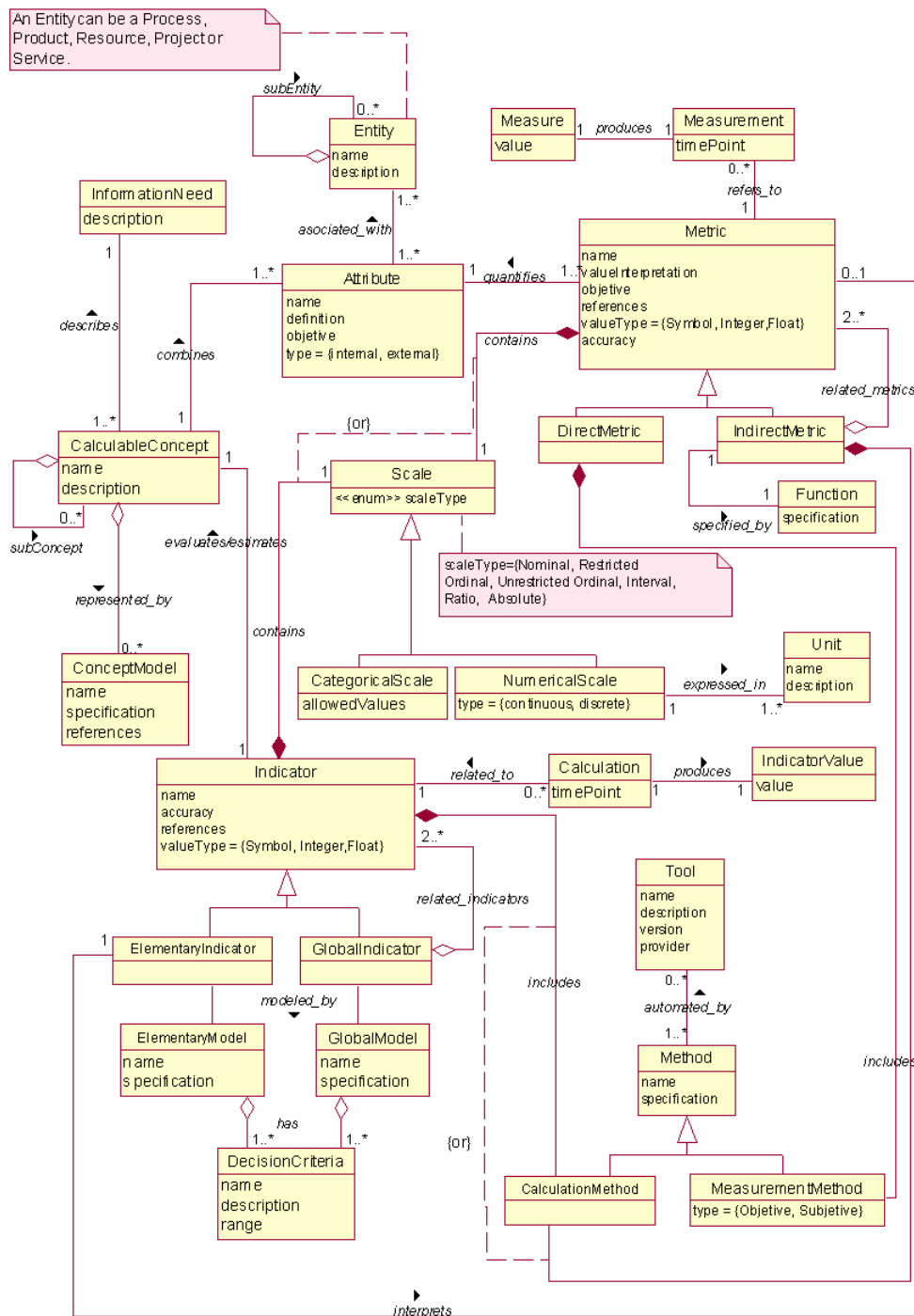


Figura 2.17: Diagrama UML que muestra los conceptos y relaciones de la ontología de medición y evaluación de [Martín & Olsina, 2003].

para ésta última, y un método de medición (para métricas directas) o de cálculo (para métricas indirectas), incluyendo las herramientas que los automatizan. También pueden especificarse los resultados de aplicar las métricas registrando los metadatos de la acción de medición y la medida resultante;

- Diseño e implementación de la evaluación: involucra la definición de indicadores (elementales y globales) para interpretar los elementos de un modelo de conceptos, utilizando un modelo de análisis acorde (elemental o global) con criterios de decisión asociados. Al igual que las métricas, los indicadores incluyen una escala y posiblemente una unidad para expresar sus valores. La acción

de cálculo también es especificada junto a los valores de indicador resultantes.

Esta ontología fue utilizada luego por los mismos autores para especificar un marco conceptual de medición y evaluación de software llamado INCAMI (siglas de *Information Need, Concept model, Attribute, Metric and Indicator*) que define los metadatos de los conceptos definidos en la ontología –entre otros- de forma estructurada [Olsina *et al.*, 2005, 2008b]. El marco está pensado para ser utilizado en la especificación y registro de los metadatos y datos involucrados en las actividades de medición y evaluación en organizaciones y permitiría asegurar que los valores medidos e interpretados sean repetibles y comparables entre los diferentes proyectos de una organización. Este marco ha sido utilizado en diferentes casos de evaluación de calidad de aplicaciones web [Covella & Olsina, 2006; Olsina *et al.*, 2006, 2008a; Lew *et al.*, 2011].

En [García *et al.*, 2004] se intenta consensuar una ontología de medición de software entre diferentes investigadores del área, incluyendo a los autores de [Martín & Olsina, 2003; Olsina & Martín, 2004], consiguiendo un consenso parcial de los términos y relaciones. La propuesta está basada en cuatro conceptos principales: (i) la forma de medir, (ii) la acción de medir (el método de medición, la función de cálculo o un modelo de análisis), (iii) el resultado de la medición (medida) y (iv) el concepto de métrica (la forma de medir y la escala/unidad en la que se expresan los valores medidos).

Luego, en [García *et al.*, 2006] se continúa con la propuesta de [García *et al.*, 2004] y se propone una ontología de medición de software (ver Figura 2.18) desarrollada a partir de la comparación y análisis de los diferentes estándares y propuestas relacionadas. La ontología está alineada con el vocabulario de metrología utilizado en otras disciplinas más robustas de la ingeniería. La ontología está organizada a su vez en cuatro subontologías:

- Caracterización y objetivos de medición de software, incluye los conceptos requeridos para establecer el alcance y objetivos de medición. Incluye la necesidad de información que identifica las entidades (pertenecientes a una clase de entidad) y los atributos a medir de estas entidades. Los atributos están relacionados a la necesidad de información mediante conceptos medibles en un modelo de calidad.
- Medidas de software, que relacionan un enfoque, una escala de medición (perteneciente a un tipo de escala) y una unidad de medición. Se definen tres tipos de medidas: base, derivadas e indicadores.
- Enfoques de medición, representan los enfoques usados por los diferentes tipos de medidas para obtener resultados. Una medida base usa un método de medición, una medida derivada usa una función de medición (basada en otras medidas bases y/o derivadas) y un indicador usa un modelo de análisis (basado en un criterio de decisión).
- Medición, establece la terminología relacionada al acto de medir. Una medición (que es una acción) es un conjunto de resultados de medición, para un atributo dado de una entidad, usando un enfoque de medición. Los resultados de medición surgen al efectuar acciones de medición.

En ontología presentada en esta última publicación, a diferencia de la presentada en [Martín & Olsina, 2003; Olsina & Martín, 2004], no se especifican los atributos o propiedades de los conceptos involucrados, es decir que la especificación de los términos no es explícita, dificultando la repetitividad y consistencia de los resultados.

Los conceptos y relaciones enunciados en la ontología presentada en [García *et al.*, 2006] fueron utilizados luego en [García *et al.*, 2007] para desarrollar un meta-modelo de medición (el mismo modelo visto en la Figura 2.18). El uso de un meta-modelo permite representar los metadatos asociados con los valores obtenidos durante la medición. El meta-modelo puede ser utilizado por las organizaciones como referencia en la definición de sus modelos de medición, permitiendo que todos sus datos sean

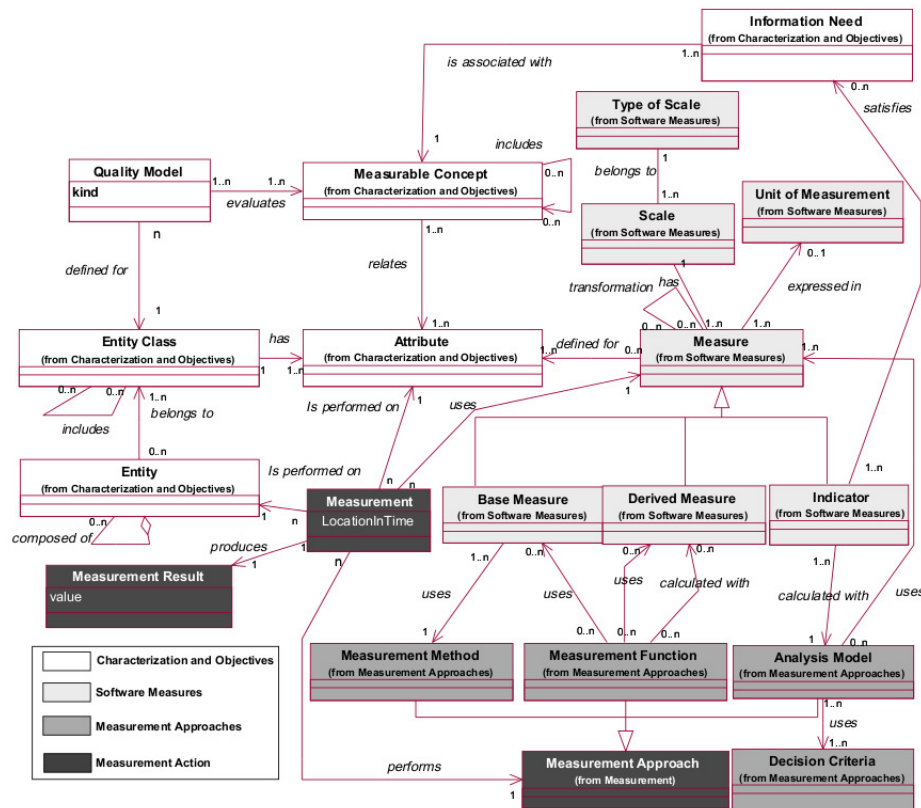


Figura 2.18: Diagrama UML que muestra los conceptos y relaciones de la ontología de medición de [García *et al.*, 2006].

representados de forma homogénea; además puede ser utilizado como base para el desarrollo de repositorios de mediciones. La ontología propuesta en [García *et al.*, 2006] también fue aplicada a tres casos de estudio [García *et al.*, 2009] para comparar y analizar estándares de M&E existentes, para definir un modelo de calidad para un portal web de datos y para el desarrollo de un metamodelo de medición, y sintaxis gráfica correspondiente, para representar modelos de medición de software.

En [Barcellos *et al.*, 2010] se presenta una *Ontología de Medición de Software* que pretende establecer una ontología de referencia para la descripción clara y precisa de las entidades utilizadas en la medición de software con propósitos de comunicación, aprendizaje y resolución de problemas. Los autores acuerdan que dicha ontología debe ser construida utilizando un lenguaje bien fundado ontológicamente, para lo cual utilizan un meta-modelo (definido como un perfil UML) que representa los elementos de una ontología fundacional¹⁶. Los autores fundamentan su propuesta argumentando que el dominio aún no ha sido bien definido por ser una disciplina relativamente joven y que las propuestas existentes no utilizan una ontología fundacional como base sino que se apoyan en modelos de baja expresividad. La ontología propuesta se estructura en 5 subontologías, cada una incluyendo un conjunto de conceptos y relaciones. La principal subontología, *Entidades Medibles y Medidas*, incluye las entidades objeto de medición, sus propiedades (elementos medibles) y las medidas usadas para medirlas (ver Figura 2.19). La subontología *Objetivos de Medición* cubre la alineación de medición con los objetivos organizacionales. La subontología de *Definición Operacional de Medidas* cubre la definición de los aspectos relacionados a la recolección y análisis de medidas establecido por una organización de acuerdo a sus objetivos de medición. La subontología de *Medición de Software* se refiere a la recolección y almacenamiento de los datos de las medidas. La subontología de *Resultados de Medición* maneja el análisis de los datos recolectados para obtener información que soporte la toma de decisiones.

¹⁶ *Unified Foundational Ontology* desarrollada por Giancarlo Guizzardi en su tesis doctoral (2005).

Finalmente, la subontología de *Comportamiento del Proceso de Software* se refiere a la aplicación de los resultados de medición al análisis del comportamiento de los procesos de software organizacionales. Adicionalmente, estas subontologías se relacionan con ontologías del dominio de la organización y del proceso de software. No obstante, la base ontológica del modelo no incluye las propiedades que caracterizan a cada uno de los conceptos, con lo cual la especificación del dominio resulta imprecisa para ser utilizada en la medición de software en proyectos concretos de una organización.

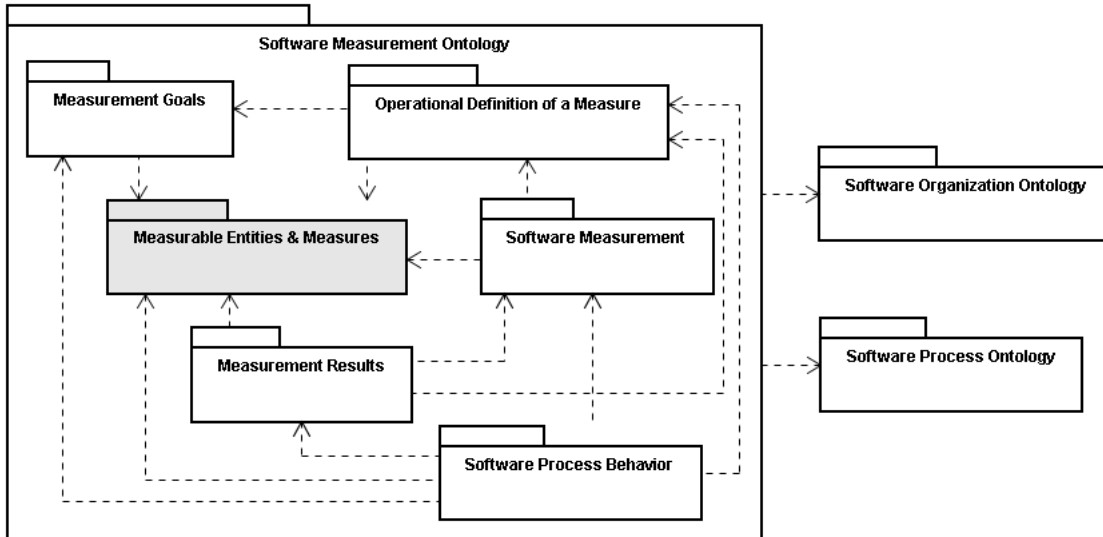


Figura 2.19: Ontología de medición de [Barcellos et al., 2010].

Adicionalmente otras fuentes de la literatura [Zuse, 1998; Kan, 2002; IEEE Std. 1061, 2004; Laird & Brennan, 2006] describen los elementos a incluir en la especificación de métricas y el reporte de mediciones aunque no presentan un modelo estructurado que incluya los elementos de medición enumerados en 2.1.4.

2.1.6. Necesidad de información de contexto en medición y evaluación.

Como se indicó en la sección 2.1.4, uno de los elementos a incluir en una especificación de medición y evaluación es la descripción o caracterización del entorno o contexto de la entidad a medir, relevante al objetivo de medición, es decir aquellos factores de la organización, proyecto, procesos, recursos y otras entidades relacionadas que influyen en el objetivo de medición. Contar con esta información de forma clara y estructurada es particularmente importante en estrategias de medición orientadas a objetivos donde la necesidad de información guía todas las actividades posteriores, es decir, diseño e implementación de la medición y la evaluación, y posteriormente la interpretación y toma de decisiones. Pasar por alto esta información podría llevar a una implementación inapropiada de la medición, interpretaciones incoherentes de tales mediciones, conclusiones erróneas y consecuentemente, a una toma de decisiones que alejen a la organización de los objetivos originales [Kitchenham et al., 2007].

La necesidad de conocer la información del contexto relevante en actividades de medición y evaluación ha sido expresada previamente en varios trabajos de la literatura relacionada, como se muestra a continuación, aunque ninguno de ellos provee un modelo estructurado que permita especificar de manera clara dicha información. Cabe aclarar que diversas fuentes utilizan el término “*environment*” –que aquí se traducirá como “entorno”– para referirse al conjunto de aspectos o factores que afectan o influyen en la medición y evaluación de entidades y al que esta tesis se refiere como “*contexto*”. En la sección 2.2 se revisan las definiciones del término “*contexto*” que proponen diversos autores .

En el estándar ISO/IEC 15939 [ISO/IEC 15939, 2001] se contempla la especificación de información

de contexto relevante para las tareas de medición y evaluación, así como su importancia, al describir la actividad *Caracterizar la unidad organizacional*, como parte de la actividad *Planificar la Medición*:

“Las características de la unidad organizacional que son relevantes para seleccionar las medidas e interpretar los productos de información deben ser explícitamente descriptas. La unidad organizacional provee el contexto de medición, y por lo tanto es importante hacer explícito este contexto y las suposiciones que representan y las restricciones que impone.”

Luego indica potenciales fuentes de información de contexto al enunciar que “la caracterización puede especificarse en términos de procesos organizacionales, dominios de aplicación, tecnología, interfaces entre las divisiones/departamentos y estructura organizacional.” Luego recomienda continuar utilizando la información de contexto especificada “en todas las actividades y tareas subsecuentes”. El estándar también indica que el contexto necesario debe ser almacenado cuando se recolectan datos, durante la actividad *Realizar el proceso de medición*, para “verificar, comprender o evaluar los datos”; también durante la actividad *Analizar datos y desarrollar productos de información*, se indica que “todas las interpretaciones deben tener en cuenta el contexto de las medidas”.

En [Park *et al.*, 1996] se presenta un proceso adaptable para identificar y definir medidas de software para soportar los objetivos de la organización. En este documento, los autores destacan la importancia de los modelos mentales de los agentes interesados afirmando que estos “proveen contexto y foco”, argumentando que todos los usan para interpretar y actuar sobre las observaciones, por lo que estos modelos deben ser explicitados para ser efectivos en actividades de mejora orientadas a objetivos y, así, proveer conocimiento compartido sobre los contextos para interpretar y actuar sobre los resultados de mediciones. En este punto los autores también mencionan la importancia de los factores del entorno resaltando que “todos los procesos operan en un ambiente que contribuye o perjudica su esperanza de éxito” y enumeran un número de factores que pueden influenciar el rendimiento de los procesos, los cuales, si son cuantificados, pueden ayudar a interpretar los datos obtenidos de las entidades bajo análisis. Los autores también proveen actividades guía para especificar la información de contexto. Durante la identificación de atributos a medir se indica incluir “aquellos que, si son cuantificados, ayudan a responder las preguntas o establecer un contexto para interpretar las respuestas”. Luego, cuando se formalizan los objetivos de medición, los autores describen un campo reservado para hacer explícito el entorno que incluye “todas las restricciones significativas sobre objeto medido (tiempo, recursos, criterios de rendimiento inusual, etc.), así como restricciones sobre el alcance o tiempo del proceso de medición mismo”, resaltando que si “el contexto no es explicitado, puede no ser entendido por todos los que usen los datos reportados”, por lo que “las probabilidades de uso indebido de los mismos son altas y se puede llegar fácilmente a conclusiones erróneas”. También se proveen ejemplos de descripciones de entornos, aunque utilizando un formulario o plantilla donde el contexto es descrito de forma textual. Ya que los autores argumentan que el contexto es abierto, se proveen recomendaciones de cómo determinar los límites o alcance del mismo en función de las entidades (y sus atributos pertinentes) que afectan a la entidad siendo medida y analizada.

En [Bailey & Basili, 1981], se propone un proceso de generación de modelos de predicción de costos para proyectos de desarrollo de software que captura los factores contextuales particulares de una organización y las diferencias entre los proyectos, obteniendo como resultado modelos adaptados a la misma. El proceso propuesto incluye una actividad donde los datos del entorno de programación de cada proyecto son recolectados para proveer una fundamentación del esfuerzo y los recursos necesarios por los proyectos de software. Los autores proveen como ejemplos de estos factores las “*metodologías usadas durante el diseño y el desarrollo, experiencia del cliente y de los programadores*”, por mencionar algunos. Además se muestran resultados de aplicar el proceso usando datos recolectados del *Laboratorio de Ingeniería de Software del NASA/Goddard Space Flight Center*. Los autores concluyen que el desafío principal es determinar “qué atributos del contexto realmente capturan la razón de las diferencias” en los aspectos estudiados entre los diferentes proyectos.

También en el enfoque *Goal-Question-Metric* [Basili *et al.*, 1994] el contexto es considerado a nivel conceptual cuando se expresa que “un objetivo es definido para un objeto, por una variedad de razones, con respecto a varios modelos de calidad, desde varios puntos de vista, relativo a *un entorno particular*”, aunque en la plantilla presentada en dicho trabajo no se incluye campos asociados a una descripción de dicho entorno.

En [Basili & Rombach, 1988], donde se presenta un modelo de proceso orientado a la mejora basado en el paradigma GQM (como parte del proyecto TAME – *Tailoring A Measurement Environment*), se reconoce desde el principio que un proyecto de software está embebido en algún entorno, caracterizado por un conjunto de propiedades las cuales –como sostienen los autores– pueden ser caracterizadas por métricas objetivas o subjetivas. Los autores extraen un conjunto de principios a partir de lecciones aprendidas de medir y evaluar procesos y productos de Ingeniería de Software en una variedad de entornos de proyectos. Algunos de estos principios, que se resumen a continuación, reflejan la necesidad de adaptar “los procesos de planificación y análisis debido a necesidades cambiantes entre diferentes proyectos y entornos”:

- Todos los entornos de proyectos y productos son diferentes de alguna forma; estas diferencias deben ser explicitadas y tomadas en cuenta en los procesos de ejecución de software y en los objetivos de calidad del producto.
- Los modelos de ejecución a utilizar deben estar adaptados a las necesidades y características de la organización y del proyecto.
- La adaptación de procesos necesita ser formalizada teniendo en cuenta (entre otros) las características del entorno del proyecto y de la organización, ya que es difícil aplicar de forma abstracta métodos definidos para entornos específicos.
- Debido a las diferencias entre los modelos de ejecución, los modelos y métricas son dependientes del entorno, por lo que deben ser adaptadas y validadas para el entorno en el cual serán aplicadas.
- Para permitir la interpretación y el análisis contextual válido, el proceso de medición, debe ser *top-down* en lugar de *bottom-up*.
- Para cada entorno existe un conjunto característico de métricas que provee la información necesaria con propósitos de interpretación.
- Las caracterizaciones de los entornos locales deben ser almacenados en bases de experiencias históricas para evaluar y comparar los proyectos.
- Las métricas deben estar asociadas con interpretaciones que deben ser dadas en contexto.

La descripción del modelo de proceso presentado incluye la caracterización formal del “estado actual del entorno de un proyecto”, necesaria “para entender los varios factores que influyen el entorno del proyecto actual”, los cuales también pueden ser analizados en base a datos de proyectos anteriores, estableciendo así un punto de partida para la mejora. Durante la planificación, se definen los objetivos siguiendo el paradigma GQM, incluyendo al entorno que consiste de “factores de procesos, factores de personas, factores de problemas, métodos, herramientas, restricciones, etc.” Luego, los objetivos son refinados en preguntas cuantificables que incluyen, entre otros:

- *Preguntas relacionadas a los productos*, incluyen, entre otros, preguntas relacionadas al contexto del mismo, descrito como “una caracterización cuantitativa de la comunidad de clientes que usan este producto y sus perfiles operacionales”. También incluyen una perspectiva de *calidad de interés*, constituida por, entre otros, preguntas relacionadas a la “validez del modelo para el entorno particular”, esto es, “un análisis de qué tan apropiado es el modelo para el entorno particular del proyecto”.

- Preguntas relacionadas al proceso, incluye la perspectiva de calidad de interés, de la misma forma que en las preguntas relacionadas al producto.

Basili & Rombach también describen una *Base de Experiencias* incluyendo un número de dimensiones entre las cuales se incluye “el grado en el que se encuentra adaptada para satisfacer las necesidades específicas del proyecto (contexto)”. Esta dimensión de contexto permite registrar los datos acerca de un proyecto específico, grupo de proyectos, procesos o productos, proporcionando la habilidad de generalizar experiencias a contextos más amplios. Finalmente, la caracterización del entorno es incluida entre los elementos destinados a la mejora del proceso.

En [Gresse *et al.*, 1995] también se utiliza el enfoque GQM como base para un modelo de proceso para programas de medición y, como tal, utiliza la misma base conceptual del caso anterior. Según proponen los autores, la definición del proceso comienza por caracterizar “el proyecto actual y su entorno respecto de modelos y métricas existentes.” También afirman que los programas de medición basados en GQM deben ser configurados y ejecutados de acuerdo a tres principios, el primero de los cuales enfatiza en mantener los objetivos de medición explícitos (y en consecuencia, la descripción del entorno). Este énfasis también es hecho a nivel operacional en el principio que enuncia que “cada métrica debe tener una fundamentación subyacente que es documentada de forma explícita usada para justificar la recolección de datos y para guiar el análisis e interpretación de los mismos”. Además, cuando se describe el contenido de las plantillas para los objetivos de medición, particularmente para el entorno del análisis, los autores dan ejemplos tales como “la organización, el proyecto, el modelo de proceso usado”. El uso de información de contexto también se hace evidente en las etapas del proceso propuesto. Durante la actividad de pre-estudio se actualiza (entre otros) la “*descripción del entorno*” que incluye:

- Lista de entradas disponibles, condiciones y restricciones: información disponible en la organización, relevante a la introducción del programa de medición y documentación explícita de información relevante al programa de medición no documentada explícitamente.
- Caracterización organizacional: describe el entorno (contexto organizacional) donde se realiza la medición. Incluye la organización misma, sus modelos de proceso, modelos de producto, modelos de calidad y prácticas de recolección de datos.
- Objetivos de mejora organizacional: lista los problemas existentes y los objetivos de mejora resultantes de toda la organización.
- Caracterización del proyecto: problemas conocidos, modelos de calidad, procesos y recursos del proyecto donde se lleva a cabo la medición (incluye la instanciación del proceso organizacional y la descripción del producto específico).
- Objetivos del proyecto: especifica los objetivos del proyecto específico.
- Plan del proyecto: descripción de los aspectos técnicos y de gestión de un proyecto (por ejemplo, tareas a realizar, recursos requeridos, métodos y procedimientos, cronograma, etc.)

Durante el pre-estudio se utiliza documentación de programas de medición existentes como entrada provenientes de “programas realizados en el mismo contexto o en un contexto comparable”, aunque los autores no especifican cómo se determina esta similitud de contexto.

En [Briand *et al.*, 2002], donde se propone un enfoque “para definir medidas de atributos de producto en Ingeniería de Software”, los autores argumentan que uno de los problemas más relevantes relacionados con la validez teórica y empírica de las medidas es que sus “definiciones no siempre tienen en cuenta el entorno o contexto en el cual fueron aplicadas”. En el enfoque presentado, los autores señalan que “la información específica del entorno proveniente de los equipos de proyecto y de la fábrica

de experiencias se encuentra disponible para casi toda actividad”, particularmente en el primer paso de alto nivel, donde los objetivos corporativos son refinados en objetivos de medición. Aquí los autores afirman que “las dimensiones de los objetivos tienen un impacto directo en los pasos restantes del enfoque de definición de medidas y del programa de recolección de datos”; particularmente, que el entorno ayuda a determinar (i) “el contexto en el cual el estudio es llevado a cabo” y (ii) “el alcance dentro del cual los resultados del estudio son válidos y pueden ser generalizados”.

El uso de información de contexto también es evidente durante la definición de medidas, ya que una nueva medida es definida o una existente es seleccionada sólo si satisface los valores de un conjunto de propiedades del entorno bajo estudio.

También se presenta un modelo conceptual (mostrado en la Figura 2.16), en el cual, un *Objetivo de Medición (Measurement Goal)* tiene asociado, entre otros, un *Entorno (Environment)*, aunque no se provee información adicional, tal como una definición o estructura asociada. Los autores también proponen el uso de plantillas para definir objetivos, siguiendo el enfoque GQM, donde el entorno es uno de sus componentes (incluyendo datos del equipo, del proyecto, del producto, etc.), aunque de forma no estructurada.

Respecto del uso de contexto en la interpretación de los resultados, los autores afirman que “el entorno muestra el alcance (mínimo) en el cual los resultados del estudio empírico son válidos” y si los resultados son usados en otros entornos, las similitudes y diferencias con el entorno original deben ser cuidadosamente estudiadas, aunque, nuevamente, sin describir cómo. Los autores también muestran que contar con una caracterización explícita y cuantitativa del entorno desde un principio puede ayudar a encontrar las razones de problemas detectados luego que el análisis fue realizado, y actuar de alguna forma para resolverlos.

En [Card, 2000], el autor se refiere a las entidades del entorno del negocio a las cuales las necesidades de información están relacionadas, es decir, *empresa, proceso, proyecto, producto y mercado*. La forma en que la información de contexto afecta las necesidades de información se evidencia en la siguiente afirmación: “la naturaleza del mercado en el cual los productos son ofrecidos determina el tipo y prioridad de las necesidades de información”. Finalmente, el autor reconoce la necesidad de información de contexto cuando afirma que “la gestión por hechos incluye no solo reportes de números, sino también el contexto necesario para tomar decisiones basadas en tales números”.

En [Kitchenham *et al.*, 2001], los autores reconocen que “hay conjuntos de métricas que identifican un conjunto particular de medidas apropiadas para un contexto particular”. Los autores también tratan los problemas que surgen cuando la información de contexto no es usada, al afirmar que “los problemas de validación, almacenamiento y análisis de datos surgen de definiciones pobres, pero también cuando hay información de contexto ausente relacionada a los objetivos de medición”. Los autores también cubren la importancia de conocer el contexto durante la definición de las métricas, por ejemplo, cuando se determina “si se requiere una definición completa de cada punto en una escala en un contexto particular”; los autores proveen guías para este escenario.

La necesidad de información de contexto es aún más evidente en evaluaciones de calidad en uso, como se expone en [Bevan, 1999] donde se declara que “se debe tener cuidado en la generalización de resultados de cualquier medición de *calidad en uso* a otros contextos con diferentes tipos de usuarios, tareas o entornos”. Las caracterizaciones de usuarios, sus objetivos, y el contexto de uso relevante son necesarios cuando se especifica y mide calidad en uso.

En [Florac *et al.*, 1997], los autores presentan una guía, basada en sus propias experiencias y de otros, que describe e ilustra las mejores prácticas para recolectar y usar datos cuantitativos en la gestión y mejora de procesos de software. Según los autores, las medidas de proceso son dirigidas por objetivos de negocio y técnicos que, según afirman, deben estar formulados y bien estructurados –deben incluir un número de elementos entre los cuales se incluyen la descripción del entorno y las restricciones que, según los autores “provee un contexto para interpretar definiciones de mediciones y resultados. Cuando el contexto no es hecho explícito, puede no ser comprendido por todos los que recolectan o usan los

datos reportados. Las probabilidades de mal uso de los datos son, entonces, importantes, ya que es muy fácil asumir de forma implícita cosas que no son así y, por lo tanto, llegar a conclusiones erróneas”.

Los autores también muestran cómo las medidas para algunas de las perspectivas, que ellos consideran centrales al proceso de medición, pueden ser usadas como información de contexto para interpretar otros aspectos de la gestión de procesos. Por ejemplo, las mediciones de conformidad pueden ser usadas “para proveer información contextual que ayuda a explicar resultados de rendimiento y las variaciones y patrones que son observados”. También, cuando se analizan medidas de desempeño de procesos para evaluar la estabilidad y capacidad de los procesos, los autores argumentan que la información sobre el contexto en el cual los datos fueron producidos es requerida para ser interpretadas apropiadamente.

La importancia de los datos de contexto es tal que los autores afirman que su captura, al momento de reportar mediciones de producto y proceso, debe ser incluida en el proceso de recolección de datos. Aún más, la recolección de datos de contexto a través de la medición es considerada por los autores como un principio general a ser seguidos por las organizaciones cuando se obtienen y almacenan datos para gestión de procesos. Así mismo, los autores también tienen en cuenta los datos de contexto cuando se diseñan bases de datos, ya que proponen “capturar y retener definiciones y descripciones de contexto, no solo datos de mediciones directas” y también para “asociar valores medidos a los contextos y entornos en los que fueron recolectados (utilizando propiedades como descriptores de producto, entorno y proceso; el estado del proceso y proyecto; el tiempo y lugar de medición; el método de medición; entre otros).”

Los autores también enfatizan la importancia del contexto en la interpretación de datos al enunciar que “los datos de medición deben contener los elementos de datos esenciales y las definiciones asociadas e información contextual necesarias para comprender e interpretar los valores” ya que consideran la revisión de “resultados de medición en el contexto de los procesos y entornos que los producen” como un principio de éxito en la medición de procesos.

Todos los autores citados anteriormente acuerdan en que la información de contexto es necesaria desde la planificación de medición y evaluación de entidades de software hasta su ejecución y el análisis de los resultados correspondientes. En algunos casos se proveen guías a seguir para la identificación de aquellas propiedades del entorno que afectan el diseño, implementación e interpretación de los resultados de medición. Sin embargo, la información del contexto relevante en medición y evaluación no es útil si no se encuentra definida y estructurada de forma clara para poder ser comprendida y comunicada de forma coherente por todos los interesados. Concretamente, se deberían proveer detalles acerca de los metadatos (conceptos, atributos y relaciones) necesarios para especificar la información de contexto.

2.2. Enfoque Sensible al Contexto

El enfoque sensible al contexto, aplicado a los sistemas de información, propone incorporar, de forma complementaria a la información de dominio manipulada por un producto o proceso de software, una nueva dimensión de información que consiste en la descripción del contexto que influye o afecta la información del dominio del producto o proceso. Este enfoque se apoya en la premisa ampliamente aceptada que toda información no puede ser correctamente interpretada de forma aislada, es decir, sin considerar el contexto en el cual se encuentra embebida.

La aplicación de este enfoque implica conocer y procesar cuantitativamente las propiedades o atributos que describen al contexto y que influyen en la interpretación y procesamiento de la información del dominio. Para lograr esto es necesario cumplimentar un conjunto de requisitos ingenieriles, a saber:

1. determinar qué es contexto en el entorno del dominio de aplicación y qué información será capturada respecto del mismo (es decir su alcance);

2. diseñar un modelo de información que permita representar de forma apropiada la información de contexto relevante y la integración y contextualización de la información en aplicaciones para el dominio respectivo;
3. definir un modelo de proceso que describa los mecanismos necesarios para capturar, procesar y utilizar la información de contexto en el dominio de aplicación.

Las decisiones sobre los requisitos anteriores deben ser tomadas en consideración del propósito por el cual se aplica el enfoque. Estas decisiones determinarán la generalidad de la aplicación resultante, que sería deseable en alguna medida con fines de reusabilidad, aunque sin descuidar el equilibrio con aspectos de desempeño y aplicabilidad práctica.

2.2.1. Definiciones

Antes de poder determinar cuál es el contexto relevante para una aplicación dada es necesario entender qué es contexto, cómo se relaciona con la información específica de un dominio y qué implica para una aplicación ser *sensible al contexto*. Al respecto, en la literatura pueden encontrarse diferentes definiciones del término ya que el uso del mismo es relativamente nuevo en el ámbito de las tecnologías de información y, particularmente, de la ingeniería de software. A continuación se presentan algunas de las visiones y definiciones de contexto (desde las más generales hasta las más particulares) que se consideraran más relevantes para el propósito de esta tesis.

Lejos de una definición estructural, [Dourish, 2004] analiza la noción de contexto en el área de la computación ubicua identificando dos visiones diferentes. La primera, de origen técnico, ve al contexto como un problema de representación para conceptualizar la acción humana y las relaciones entre esa acción y los sistemas computacionales que la soportan. El autor distingue los siguientes supuestos que subyacen a esta vista:

- el contexto es relativo a algún objeto y puede ser codificado y representado en sistemas de software como cualquier otra información;
- el alcance del contexto es estable para una aplicación dada: por lo tanto es posible determinar por anticipado lo que se considera contexto y la relevancia de cada elemento que lo compone;
- la actividad y el contexto en el que se lleva a cabo son separables ya que la actividad puede cambiar de contexto.

El autor hace notar que, independientemente de su definición y aplicación, estos supuestos respaldan la noción de contexto como un conjunto de características del entorno que rodea a una actividad y que pueden ser codificadas e incorporadas a un sistema de software junto a la información de la actividad misma. Es decir, los sistemas modelan, representan y capturan el contexto al igual que los datos relativos a la actividad principal de los mismos.

La segunda visión, de origen social, ve al contexto como un problema de interacción en la transferencia de ideas entre diferentes dominios intelectuales manteniendo el significado de las mismas. El autor distingue para este caso los siguientes supuestos análogos a los mencionados arriba:

- se considera al contexto, no como información en sí misma, sino como una propiedad relacional entre objetos o actividades;
- el alcance del contexto es determinado dinámicamente así como su relevancia, dependiendo del entorno, actividad y participantes particulares;
- el contexto surge de forma activa de la actividad, y es mantenido durante su transcurso.

En contraposición a las visiones analizadas, Dourish presenta una visión alternativa de contexto en interacción. Esta visión no constituye un modelo técnico o diseño diferente de como se puede representar contexto, sino más bien es una propuesta diferente acerca de qué es contexto: el autor presenta un modelo de contexto en el cual el contexto y la actividad se constituyen mutuamente y donde la importancia de contexto no es *qué es* sino *qué hace* en la interacción –es decir, el rol que juega y las formas en que es mantenido y gestionado.

Siguiendo las definiciones de Dourish, en [Dos Santos, 2008] se presenta una definición de contexto híbrida que combina aspectos de las dos visiones presentadas por aquel –la vista de representación y la de interacción. La visión de representación se aplica al concepto de *elemento contextual*, definido como “cualquier pieza de datos o información que permite caracterizar una entidad en un dominio”. En este sentido, un elemento contextual puede ser conocido, codificado y representado así como es posible determinar por adelantado qué será considerado un elemento contextual. La vista de interacción se aplica al concepto de *contexto* de una interacción entre un agente y una aplicación para ejecutar una tarea, definido como “el conjunto de elementos contextuales instanciados que son necesarios para soportar la tarea en cuestión”. Definido de esta forma, lo que será considerado relevante en el contexto es definido dinámicamente y dependerá de una interacción o tarea en ejecución particular.

En [Theodorakis, 2001] el autor analiza las diferentes definiciones del término provistas en diferentes áreas del conocimiento (tales como psicología cognitiva y lingüística) incluyendo ciencias de la computación (tales como inteligencia artificial, desarrollo de software, bases de datos, representación de conocimiento, entre otros). Sin embargo, el autor asume una definición orientada al modelado de información donde el contexto es visto como un entorno de referencia relativo al cual se proveen descripciones de objetos del mundo real. De esta forma, el autor describe a la *contextualización* como un mecanismo de abstracción que permite representar particiones del mundo real o, en general, grupos de información, tales como situaciones, puntos de vista, espacios de trabajo o versiones. El autor también resalta la diferencia entre una situación del mundo real y un contexto: el primero es completo en el sentido que registra todos los estados del mundo, independientemente de cómo se represente en la mente del agente; el segundo es parcial ya que representa situaciones y por lo tanto captura diferentes perspectivas y niveles de detalles para una situación particular.

Siguiendo una línea más orientada a entornos operacionales, en [Dey, 2001] se define al *contexto* como “cualquier información que pueda ser utilizada para caracterizar la situación de una entidad. Una entidad es una persona, lugar u objeto que es considerado relevante en la interacción entre un usuario y una aplicación, incluyendo al usuario y la aplicación”. Además el autor define como *sistema sensible al contexto* aquel que utiliza o depende del contexto para proveer la información o servicios relevantes al usuario, donde esta relevancia depende de la tarea que realice el mismo.

En [Gong, 2005] se extiende el análisis del concepto al presentar dos interpretaciones de contexto desde un punto de vista computacional; la primera, más formal, ve al contexto como un espacio donde todos sus elementos se encuentran agrupados o contenidos por una entidad X de interés y responde a un patrón del estilo “en el contexto de X” para hacer referencia a aquellos elementos que se encuentran en el contexto de X. La segunda interpretación ve al contexto como el espacio de aquellas cosas relacionadas externamente con y referenciadas por X. Ésta interpretación responde a un patrón del estilo “el contexto de X” para hacer referencia a aquellos elementos relacionados de alguna forma con X. De estas visiones, Gong propone una definición cuantificable de contexto al definirlo como “la colección formada por un objeto y sus relaciones con otros objetos, incluyendo estos últimos”.

De forma similar, en [Huang & Tao, 2004] se define al contexto de una entidad como una colección de información situacional semántica agregando que dicha información caracteriza las propiedades u operaciones internas de la entidad y las relaciones externas bajo una situación específica. Los autores dan ejemplos de información típica de contexto incluyendo metadatos generales de entidades, anotaciones o recursos multimedia, modelos conceptuales, entre otros.

En [Strang *et al.*, 2003] se mejora el aspecto representacional de contexto al definirlo como cualquier

información que puede ser utilizada para caracterizar el estado de una entidad (persona, lugar u objeto) concerniente a un aspecto (una clasificación) específico, cuyos estados (o instancias) se expresan en una determinada escala. Además define qué es una entidad y un aspecto relevante en función de una tarea y un estado de dicho aspecto. También identifica a un sistema como sensible al contexto si éste utiliza cualquier tipo de información de contexto antes o durante el aprovisionamiento del servicio, aclarando que al conjunto de toda la información de contexto conocida lo denomina situación.

Definiciones más enfocadas en la aplicación pueden encontrarse en [Kaltz *et al.*, 2005] donde se define a contexto en el entorno de la Ingeniería Web como la combinación de todos los factores relevantes (tales como el perfil del usuario, la tarea u objetivo actual, la ubicación, tiempo y el dispositivo utilizado) con el fin de satisfacer las necesidades individuales de los usuarios (es decir de personalización de la aplicación web). El autor describe además una aplicación web sensible al contexto como aquella en la que la navegación, selección de contenidos, servicios y tipo de pantalla pueden ser adaptados de acuerdo al contexto.

En [Maalej *et al.*, 2006] se resalta la dificultad de definir de forma universal el término contexto ya que es un concepto genérico con muchas facetas que difieren de un dominio a otro. En este sentido los autores acuerdan con las visiones propuestas por Theodorakis y Dey al describir al contexto, respecto del dominio de la Ingeniería de Software, como una forma específica de conocimiento acerca del usuario y su entorno que puede ser útil para adaptar la aplicación, ya sea desde el punto de vista funcional o no funcional. Así mismo, los autores se refieren al término sensible al contexto, utilizado principalmente en el campo de la ciencia de la computación, para describir sistemas que mantienen información del contexto bajo el cual operan y ajustan su comportamiento de acuerdo a dicha información, dando como ejemplos aplicaciones de dominios de aplicación tales como servicios móviles basados en la posición y computación ubicua. Sin embargo, los autores resaltan que el enfoque sensible al contexto no es único para esos campos, dando ejemplos de aplicaciones de personalización de servicios tales como la adaptación de interfaces de usuario o el ofrecimiento de recomendaciones sensibles al contexto.

Cabe resaltar aquí la importancia de otro concepto relacionado íntimamente con el enfoque sensible al contexto, no tratado explícitamente por la mayoría de los trabajos citados: el de *contextualización*. En el mundo real el contexto siempre se encuentra presente para toda entidad bajo consideración. Sin embargo, en los sistemas de información la descripción del contexto debe ser explícitamente asociada a la información que describe una entidad del mundo real para poder ser utilizada. La *contextualización* se refiere entonces a la acción de asociar a una determinada entidad, de interés para el propósito de la aplicación particular, una descripción o caracterización del contexto de esa entidad relevante para dicho propósito, usando el formalismo de representación elegido [Maalej *et al.*, 2006]. Esta asociación impone un conjunto de condiciones sobre la entidad en el sentido que ésta es válida o aplicable bajo dichas condiciones. En resumen, no es suficiente describir el contexto, sino que éste debe ser asociado a la entidad que caracteriza.

2.2.2. Representación

El segundo requisito, como se mencionó al principio de la sección, es el de contar con un modelo para representar de forma apropiada la información de contexto relevante y la integración y contextualización de la información en aplicaciones para el dominio respectivo. En este caso también existen en la literatura diversas propuestas de modelado de información de contexto, utilizando diversos enfoques y con diferentes niveles de generalidad, apuntando a diferentes dominios de aplicación. Al respecto, y como punto de partida, en [Strang & Linnhoff-Popien, 2004] se presenta un análisis de los diferentes enfoques utilizados para representar contexto en el dominio de la computación ubicua. Los enfoques son evaluados respecto de un conjunto de requerimientos identificados por los autores para dicho dominio. Los autores identifican los siguientes enfoques de modelado: modelos de clave-valor, modelos de esquemas de etiquetas, modelos gráficos, modelos orientados a objetos, modelos basados en lógica y

modelos basados en ontologías. Los autores concluyen que el enfoque más prometedor para el modelado de contexto en entornos de computación ubicua, con respecto a los requisitos enumerados, es el enfoque de ontologías. Sin embargo no descartan que los enfoques restantes puedan ser aptos para el mismo propósito, más aun considerando que la lista de enfoques de modelado de contexto es incompleta y pueden surgir nuevos enfoques a considerar. Un aspecto que los autores no consideran es la combinación de diferentes enfoques de modelado. También en [Maalej *et al.*, 2006] se analizan diferentes enfoques para modelar y representar información de contexto. Entre ellos se consideran representaciones basadas en lógica –tales como la lógica descriptiva, de primer orden y otros modelos basados en éstas; representaciones que utilizan meta-modelos y ontologías –para definir tanto los términos relacionados al contexto como aquellos específicos del dominio de aplicación; y lenguajes descriptivos serializables –tales como RDF (*Resource Description Framework*) [W3C-RDF, 2004] y otros lenguajes de marcas como XML (*eXtensible Markup Language*) [W3C-XML, 2012] y sus derivados.

Tomando los resultados de los trabajos citados anteriormente como parámetro, se analizan a continuación un número de propuestas de modelado de contexto para analizar diferentes formas de estructurar la información de contexto e integrarla al dominio de aplicación.

En [de Freitas Bulcão Neto & da Graça Campos Pimentel, 2005; Wang *et al.*, 2004], se definen ontologías de alto nivel para un conjunto de dimensiones genéricas predefinidas con posibilidad de extenderlas para agregar conceptos específicos provenientes de ontologías de más bajo nivel de diferentes dominios de aplicación. En ambos casos, el contexto es descrito utilizando las entidades, y sus propiedades correspondientes, definidas en las ontologías propuestas. Particularmente, en [de Freitas Bulcão Neto & da Graça Campos Pimentel, 2005] el modelo de contexto se compone de cinco ontologías de alto nivel para las dimensiones *actividad*, *actor*, *ubicación*, *dispositivo* y *tiempo*, cada una de las cuales define un conjunto de propiedades de interés. En algunos casos, las ontologías de alto nivel hacen reuso de ontologías existentes para describir propiedades de contexto de interés. Por otro lado, en [Wang *et al.*, 2004] el modelo de contexto se estructura en una ontología de alto nivel, que captura propiedades generales de entidades contextuales básicas –tales como *persona*, *actividad*, *entidad computacional* y *ubicación*–, y una colección de ontologías de dominio específicas que definen los conceptos y propiedades en cada subdominio correspondiente. En ambas propuestas, las ontologías están especificadas en OWL (*Ontology Web Language*) [W3C-OWL, 2004] para aprovechar, por su condición de recomendación de la W3C, sus capacidades de interoperabilidad semántica e inferencia.

De forma similar, en [Kaltz *et al.*, 2005] la información de contexto se extrae de ontologías de dominio y se predefinen un conjunto de categorías contextuales. En dicha propuesta el contexto se representa como un conjunto de factores de contexto extraídos de ontologías de dominio y estructurados a partir de un conjunto de categorías predefinidas. Los elementos de información del dominio de aplicación son contextualizados estableciendo relaciones a los elementos del espacio del contexto (los factores de contexto). Estas relaciones se establecen en lo que los autores denominan un *espacio de relevancia*, donde cada relación indica el peso o grado de relevancia que posee el factor de contexto para el elemento de dominio de la aplicación correspondiente. Las categorías de contexto propuestas son de alto nivel –*tiempo*, *dispositivo*, *ubicación*, *proceso&tarea*, *usuario&rol*– pudiendo ser subdivididas en categorías más específicas según las necesidades. Los autores argumentan que tal categorización satisface una amplia variedad de escenarios de aplicaciones web y permite un amplio grado de reuso de catálogos de contexto. Para la representación de la información de contexto los autores proponen el uso de OWL por la posibilidad de integrar la especificación de contextos a ontologías de dominio, para lo cual los autores definen las clases y propiedades OWL correspondientes al modelo de contexto diseñado.

También en [Huang & Tao, 2004] se presenta un enfoque sensible al contexto cuyo modelo se basa en el uso de ontologías de dominio que pueden ser conectadas para ampliar los conceptos utilizados para describir el contexto. En el modelo propuesto por los autores el contexto está estructurado en múltiples niveles:

- En el nivel más bajo se describen los *objetos* a partir de los cuales se extrae la información de contexto (estos pueden ser diagramas conceptuales, documentos de texto, imágenes, etc.).
- En el siguiente nivel, dichos objetos son caracterizados por *descripciones intuitivas*, cercanas al lenguaje natural, utilizando términos, diagramas o ejemplos para mostrar las relaciones básicas entre los objetos.
- En el próximo nivel se utilizan *descripciones abstractas* acerca de las descripciones intuitivas.
- La capa superior del modelo consiste del *contexto foco*, es decir la *entidad* para la cual se describe el contexto.

Las descripciones de las dos capas intermedias componen el núcleo del conocimiento contextual. Estas descripciones se realizan mediante un conjunto de metadatos agrupados en cinco categorías: *general*, *perfil*, *técnico*, *objeto* y *anotación*. En la categoría *objeto* se definen los términos importados de ontologías de dominio. Las descripciones de contexto son serializadas utilizando lenguajes de marcas genéricos como XML y RDF para permitir su reuso actual o futuro en otros dominios de conocimiento que utilicen los mismos formalismos.

En [Kashyap & Sheth, 1997] también se utilizan ontologías de dominio para extraer las propiedades necesarias para describir el contexto, sin predefinir las dimensiones o tipos de propiedades a utilizar. Así, es posible extender el conjunto de propiedades incorporando nuevas ontologías de dominio. En el modelo propuesto por los autores, se utilizan metadatos específicos de dominio para abstraer los detalles de representación de los datos de dominios particulares y capturar el contenido de la información subyacente. El enfoque tiene dos componentes básicos: 1) metadatos específicos del dominio que abstraen los detalles de representación de los datos subyacentes del dominio; y 2) metadatos que construyen el contexto de la información del dominio e identifican los datos relevantes del mismo. Las descripciones contextuales son caracterizadas utilizando los términos (conceptos y roles) de ontologías específicas de dominio. Para la representación de contexto, los autores utilizan descriptores, llamados coordenadas contextuales. Estas son dinámicamente elegidas ya que los autores argumentan que no es posible determinar por anticipado todas las coordenadas contextuales posibles que caracterizan de forma completa la semántica del dominio de aplicación. Los autores eligen una representación parcial de contexto (basado en un sistema de lógica descriptiva¹⁷) como una colección de pares coordenada contextual-valor. Este enfoque asume que cada base de datos de dominio tiene asociada una ontología correspondiente ya que la definición de los contextos de los objetos toman sus términos y valores de esta ontología.

En todos los trabajos citados anteriormente se utilizan ontologías de dominio existentes como base conceptual para la especificación de información de contexto, lo que provee un punto de partida para la validación de tipos de datos e interpretación de la información. En los siguientes trabajos, los autores no utilizan una base conceptual para la especificación de contexto, o utilizan una base conceptual genérica para el modelado de dominios, por lo que se requiere definir en el modelo los metadatos que permitan especificar tal información.

En [David *et al.*, 2007] se utiliza un modelo orientado a objetos, utilizando UML, para describir el modelo de contexto (así como el resto de la información del sistema). En esta propuesta el contexto se especifica como una forma especial de conocimiento y se estructura como una agregación de conjuntos de propiedades de alguna entidad que se desea observar para contextualizar o “anotar” conocimiento. Las propiedades de contexto son tomadas de ontologías de dominio predefinidas por lo que no se prevén mecanismos de adaptación o extensión.

En [Dos Santos, 2008] se presenta un meta-modelo independiente del dominio para soportar la creación de modelos de contexto. El meta-modelo está definido como una extensión de UML 2.0,

¹⁷Un modelo de datos estructural para objetos llamado CLASSIC.

utilizando MOF (MetaObject Facility) [OMG-MOF, 2012] como base conceptual. Estructuralmente, el *contexto* es representado como un conjunto de *elementos contextuales* que caracterizan el contexto de alguna *entidad*. El meta-modelo define los bloques de construcción para crear modelos de contexto para una aplicación dada. Estos bloques de construcción se encuentran organizados en dos módulos:

- un modelo estructural de contexto que define *elementos contextuales* –aquellos utilizados para caracterizar el contexto de alguna entidad, las *entidades* que los contienen, relaciones entre ellos y la forma de adquirirlos;
- un modelo de comportamiento de contexto que define cómo debe reaccionar el sistema ante contextos específicos, identificando el conjunto de condiciones que determinan la variación de comportamiento del sistema.

De esta forma, para desarrollar una aplicación sensible al contexto se debe utilizar el meta-modelo como lenguaje para modelar el dominio de aplicación, determinando qué conceptos serán utilizados como elementos contextuales, como entidades a contextualizar, entre otros.

En [Belotti *et al.*, 2004] se presenta un modelo de contexto general utilizando un formalismo de modelado conceptual de datos orientado a objetos¹⁸. El concepto central del modelo es *contexto* que representa a la información de contexto que caracteriza a una *entidad* de la aplicación. El modelo presenta un sistema de tipos que provee una solución general y flexible para el intercambio de información de contexto. Se definen tipos para tres dominios:

- tipos básicos que definen valores primitivos, tales como enteros, reales, etc. así como colecciones de los mismos y jerarquías de restricciones,
- tipos que definen entidades del dominio de aplicación, tales como usuario o ubicación para aplicaciones en el dominio de la computación ubicua,
- tipos que definen tipos de contextos correspondientes a los contextos que caracterizan a un determinado tipo de entidad del dominio de la aplicación. Cada tipo de contexto esta compuesto por un conjunto de definiciones de atributos, los cuales pueden ser de cualquiera de los tipos mencionados y estar a su vez clasificados semánticamente.

Los autores resaltan que este sistema de tipos permite describir propiedades de contexto tanto con valores simples como con elementos complejos específicos de la aplicación. Sin embargo, el sistema de tipos del modelo presentado solo puede asegurar integridad y consistencia de tipos de datos.

En la misma línea de investigación y utilizando el mismo modelado conceptual como base, los autores presentan un modelo genérico para la representación de versiones en sistemas de información que soporta la gestión de datos sensibles al contexto [Grossniklaus & Norrie, 2007]. Este modelo permite especificar un número de *variantes* para un objeto dado, las cuales tienen asociados un conjunto de propiedades junto a sus valores correspondientes. A su vez, las variantes tiene asociados un conjunto de *revisiones* que apuntan a las instancias que contienen los datos reales del objeto. Las variantes de un objeto, y sus pares propiedad/valor, representan los diferentes contextos asociados al mismo.

En [Strang *et al.*, 2003] se propone un modelo de contexto, descrito mediante una representación gráfica genérica que incluye conceptos y predicados entre los mismos (que indican su cardinalidad), que luego se utiliza como base para definir una ontología de contexto, llamada CoOL (*Context Ontology Language*), utilizando OWL como lenguaje de representación. El modelo propuesto por los autores se conforma de tres conceptos principales: Aspecto, Escala y Contexto. Un *contexto* o *información de contexto* es definido como una pieza o elemento de información utilizado para caracterizar el estado o situación de una entidad respecto de un determinado aspecto. En el modelo, cada instancia de

¹⁸Llamado *Object Model*, de los mismos autores.

información de contexto tiene asociada la entidad a la cual caracteriza más un conjunto de *escalas* (con unidades asociadas) que definen el rango de valores válidos para dicha información de contexto. Cada escala representa un tipo de información correspondiente a un conjunto de *aspectos*. Un aspecto es una dimensión de la situación que describe a un conjunto de elementos de información de un mismo tipo semántico (por ejemplo coordenada geográficas o temperaturas). Además cada información de contexto tiene asociados un número de instancias de información de contexto a modo de metadatos que describen diferentes propiedades de la misma (tales como valores de calidad, tiempo de captura, etc.).

En [Lonsdale & Beale, 2004] se presenta un modelo para describir información de contexto genérica con información histórica. El modelo propuesto permite describir los elementos de contexto relevantes para una aplicación y sigue una estructura jerárquica de cuatro niveles de agregación:

- el *contexto*, que incluye toda la información de contexto disponible a lo largo del tiempo;
- un *estado de contexto*, que contiene todos los elementos actualmente presentes en el proceso actual respecto de un foco particular de aprendizaje;
- un *subestado de contexto*, que contiene el conjunto de elementos del estado del contexto directamente relevantes con el foco de aprendizaje y de la aplicación actual;
- una *característica de contexto*, que representa un elemento atómico que describe un ítem de información de contexto específico acerca del aprendiz o su configuración.

Los autores proponen además un esquema de metadatos para describir las características de contexto en términos de su *nombre*, *tipo*, *valores actuales* y *permitidos*, *fuentes*, *categoría*, entre otros. Las características de contexto se encuentran agrupadas en objetos de contexto, para los cuales se especifica el momento de su registración para proveer el aspecto dinámico del modelo. En esta propuesta las características utilizadas para describir el contexto son decididas por los creadores de contenidos y los diferentes agentes del sistema, pudiendo ser modificadas por los usuarios del sistema. El esquema diseñado es implementado en XML para poder especificar, almacenar y transferir datos de contextos a partir del modelo propuesto.

En [Schwarz, 2005] los autores utilizan RDF/S (RDF Schema) [W3C-RDFS, 2004] para definir un modelo de contexto cuya clase principal es *Context*. El contexto comprende una variedad de *aspectos* propios del dominio tales como *Aspectos Informativos* (documentos utilizados, dominios relevantes), *Aspectos Operacionales* (aplicaciones y servicios utilizados recientemente), *Aspectos Organizacionales* (rol del usuario, proyectos y departamentos en los que trabaja), *Aspectos Ambientales* (posición, hardware utilizado), entre otros. Cada uno de estos aspectos se compone de *elementos contextuales* específicos que contienen la información de contexto propiamente dicha y describen entidades del mundo real de los usuarios. Los elementos contextuales contienen además información acerca de su confianza o calidad. Una instancia de la clase *Context* representa el *contexto real* del usuario. Como el dominio de aplicación está claramente definido, tanto los aspectos como los elementos contextuales (y los datos que los describen) son definidos de antemano, sin posibilidades de ser adaptados a necesidades particulares. En este caso, la validación de tipos de datos es provista por el sistema de tipos de RDF.

También en [Klemke, 2002] se presenta un modelo de contexto sin base conceptual y un conjunto de dimensiones previamente establecidas. En este modelo el contexto se estructura como un conjunto de atributos o propiedades, cada una de las cuales corresponde a una de las dimensiones de contexto propuestas por los autores, aclarando que las dimensiones utilizadas pueden variar para cada aplicación. Las dimensiones identificadas son: *contexto del dominio*, *persona*, *tarea*, *tiempo* y *ubicación*. La dimensión *contexto del dominio* provee un conjunto de propiedades que permiten modelar dominios complejos, mediante la especificación de términos, conceptos y categorías relevantes, propias del dominio de la organización. El resto de las dimensiones básicas están definidas por un conjunto de atributos

o propiedades que pueden contener valores primitivos o complejos (como por ejemplo, categorías de una jerarquía). Los autores argumentan que, mediante el refinamiento ontológico de estas dimensiones, es posible cubrir todos los aspectos contextuales requeridos para cada aplicación. No obstante, el contenido de cada dimensión está definido utilizando un formalismo *ad hoc*, similar al de Backus-Naur, sin proveer mecanismos para la especificación de tipos de datos. Además los autores no indican de forma explícita el lenguaje utilizado para representar la información del modelo de contexto.

2.2.3. Procesamiento

El tercer y último requisito para la aplicación del enfoque sensible al contexto, como se introdujo al principio de la sección, es contar con un modelo de proceso que describa los mecanismos necesarios para manipular apropiadamente la información de contexto para satisfacer el propósito previsto para el dominio de aplicación. Las diversas propuestas que se encuentran en la literatura, algunas de ellas citadas anteriormente, proponen, junto a un modelo de contexto, los mecanismos necesarios para utilizar dicha información para proveer la funcionalidad sensible al contexto pensada para cada dominio de aplicación en particular. A pesar de las diferencias obvias entre estas propuestas, dependientes principalmente del propósito del enfoque, del modo de representación de información de contexto y de la arquitectura diseñada para soportarlo, en todos los casos resuelven tres procesos básicos respecto de la manipulación del contexto [Maalej *et al.*, 2006]: *(i)* la elicitación de la información de contexto a partir de las fuentes correspondientes, que incluye su adaptación al formato definido por el modelo de contexto diseñado, *(ii)* el procesamiento de la información de contexto necesario para tomar decisiones y cumplimentar la o las funcionalidades propuestas por la aplicación del enfoque, y *(iii)* la aplicación efectiva de los cambios o funcionalidades dependientes del contexto en función de los resultados del proceso anterior. A continuación se describen los diferentes aspectos de estos pasos, dejando en claro que algunos de ellos pueden no ser aplicables dependiendo del dominio y del propósito particular de la aplicación del enfoque:

1. Elicitación

- **Registración:** La información de contexto relevante debe primero ser registrada para luego ser actualizada cuando corresponda. La registración puede hacerse previamente bajo demanda (posiblemente siguiendo guías apropiadamente diseñadas para un conjunto de escenarios previstos) o mediante el descubrimiento dinámico de propiedades del contexto.
- **Adquisición:** la información relevante del contexto puede ser capturada usando dos enfoques alternativos [Dourish, 2004]: puede ser especificada y enlazada en algún momento a la información del dominio de aplicación para ser usada luego como clave para su recuperación, o puede ser actualizada periódicamente de forma activa. La elección entre estos enfoques responde a diferentes necesidades dependiendo del propósito de la aplicación. Por otro lado, la captura puede realizarse utilizando, o no, dispositivos o sensores para su lectura desde la fuente; estos sensores pueden ser físicos o lógicos (cuando la fuente de la información sea un componente de software) o ambos.
- **Pre-procesamiento:** los valores de la información de contexto son controlados (por su validez y coherencia) para mejorar su calidad, siempre que existan metadatos contra los cuales validar.
- **Construcción:** la información de contexto es formateada según el modelo de datos del contexto, utilizando alguna función apropiada tal como (la lista no es exhaustiva):
 - **Transformación:** los valores leídos de la fuente son convertidos a una nueva escala, por ejemplo, una cantidad de errores por módulo de software entre 5 y 10 es transformado a un valor de criticidad “medio” en una escala categórica.
 - **Agregación:** dos o más valores son agregados para producir información de contexto más general, por ejemplo tasa de errores o esfuerzo.
- **Comunicación:** la información de contexto es comunicada a los componentes que la utilizarán para proveer la funcionalidad sensible al contexto, para lo cual se consideran dos posibles modos [Maalej *et al.*, 2006; Stöttinger, 2004]: en modo “*push*” la información de contexto es enviada a los componentes interesados sin ser solicitados; en el modo “*pull*” la información es solicitada por los componentes interesados bajo demanda. Estos modos asumen la existencia de un componente intermediario encargado de entregar la información según el modo elegido.

2. Procesamiento

- **Interpretación:** La interpretación es necesaria cuando el proveedor y consumidor de la información de contexto son diferentes. La interpretación está basada en información explícita del dominio con capacidades semánticas, tales como las ontologías.
- **Comparación:** la información de contexto recuperada es comparada contra la caracterización del contexto del cual la funcionalidad de la aplicación es dependiente. En [Cruzes *et al.*, 2007; Grossniklaus & Norrie, 2007; Kessler, 2007] se presentan algoritmos que pueden ser utilizados para la comparación de descripciones de contextos.

3. **Aplicación:** los resultados de la comparación son utilizados para determinar las acciones a seguir, en base a un conjunto preestablecido de escenarios, para la funcionalidad particular del dominio y la aplicación, tales como la adaptación de interfaces de usuario o el descubrimiento y ofrecimiento de servicios.

Los tres procesos generales descritos anteriormente deben ser cubiertos para cualquier aplicación del enfoque sensible al contexto. No obstante, el detalle correspondiente a los subprocesos descritos puede variar para cada aplicación. Por ejemplo, la *agregación* de propiedades de contexto (*elicitación* > *construcción*) puede no ser necesaria para el caso de ciertas aplicaciones ubicuas donde sólo se manejen propiedades básicas tales como posición, tiempo, actividad, etc. Por otro lado, en una aplicación ubicua destinada sólo al uso dentro de una organización, el *pre-procesamiento* (*elicitación*) podría llevarse a cabo previamente al uso de la aplicación, cuando los valores de las propiedades de contexto sean actualizados en repositorios de la organización mediante rutinas que aseguran su validez. En cualquier caso, estos procesos y subprocesos deben ser considerados para cada aplicación y analizados en función de las necesidades de la misma.

2.2.4. Aplicaciones

Hasta aquí se han presentado propuestas alternativas, provenientes de trabajos relacionados de la literatura, en cuanto a la definición, representación y procesamiento de información de contexto, dejando de lado, en todos los casos, el dominio de aplicación de las mismas. Con esto se pretende destacar que el enfoque sensible al contexto no es exclusivo de ningún dominio, a pesar de haber surgido como una necesidad para satisfacer las necesidades en el campo de la computación ubicua o móvil donde la importancia de las características del contexto es más evidente. Ciertamente, el enfoque sensible al contexto puede ser aplicado a cualquier dominio del conocimiento en el que las características o propiedades del contexto afecten la interpretación de información o la ejecución de servicios. En este sentido, existen diferentes formas de incorporar el contexto en una aplicación dada [Chen & Kotz, 2000; Dey, 2001]: la información de contexto se puede capturar de forma automática y adaptar el comportamiento del sistema de acuerdo al contexto capturado (contexto activo) –por ejemplo para disparar la presentación de información o la ejecución automática de un servicio–, o permitir al usuario capturar la información del contexto disponible bajo demanda para ser almacenada y utilizada luego para realizar alguna operación sensible al contexto (contexto pasivo) –por ejemplo para etiquetar información para su posterior recuperación en servicios de consulta sensibles al contexto. A continuación se presentan diversas propuestas donde el enfoque sensible al contexto es aplicado a diferentes dominios.

Como ya se mencionó, es en el área de la computación ubicua o móvil donde el enfoque sensible al contexto ha sido aplicado desde un principio. En este entorno el contexto se centra en los dispositivos móviles y en los usuarios que los utilizan. En este área existe un importante número de trabajos haciendo uso del enfoque. Tal es el caso de [Strang *et al.*, 2003] cuyo propósito es dar soporte al conocimiento e interoperabilidad de contexto durante el descubrimiento y ejecución de servicios en una arquitectura distribuida propuesta. Los autores proponen para esto un modelo de servicio de propósito general con una extensión de contexto para que las interacciones de los servicios basados en ese modelo sean sensibles al contexto. También en [de Freitas Bulcão Neto & da Graça Campos Pimentel, 2005] se desarrolla una propuesta de modelo y arquitectura aplicada al área de la computación ubicua de forma genérica, aunque describiendo su aplicación a un entorno de aprendizaje electrónico, o *e-learning*, en un campus universitario. De forma similar, en [Lonsdale & Beale, 2004] se describe la aplicación del enfoque sensible al contexto, aunque enfocada en desarrollar un sistema de *e-learning* móvil sensible al contexto. El sistema incluye el soporte para el aprendizaje colaborativo, una interfaz gráfica y presentación de contenido, opciones y servicios sensible al contexto. El enfoque sensible al contexto también ha sido aplicado en entornos industriales con el objetivo de implementar un sistema de control móvil y ubicuo para facilitar el trabajo y mejorar la seguridad del personal de mantenimiento de una planta [Stöttinger, 2004]. En este entorno, los operarios utilizan dispositivos móviles que los asisten en la tarea de activar y desactivar de forma segura diferentes instrumentos que se encuentran físicamente distribuidos en la planta.

El enfoque sensible al contexto también ha sido aplicado en áreas de la gestión de la información y el conocimiento organizacional. En estos entornos el foco del contexto se centra en las entidades utilizadas para representar el conocimiento en una organización o sus integrantes y se utiliza con fines de interpretación, análisis e intercambio de información. Tal es el caso de [Schwarz, 2005] donde se presenta un sistema sensible al contexto para asistir a un usuario en su trabajo, consistente en su mayoría en buscar, leer, crear y archivar documentos. El modelo de contexto desarrollado incorpora elementos contextuales relevantes para satisfacer las necesidades de información potenciales del conocimiento del trabajador. También en el área de gestión de conocimiento, pero enfocado en la gestión del conocimiento organizacional y considerando la integración de dominios de diversas fuentes, en [Kashyap & Sheth, 1997] se ataca el problema de la heterogeneidad semántica en el entorno de Sistemas de Información Global (GIS). La solución propuesta se basa en abstraer los detalles de representación de los datos y capturar la información contenida mediante metadatos específicos de dominio. La interoperabilidad se logra estableciendo relaciones terminológicas entre las ontologías de dominio, al nivel de metadatos. Enfocado también en la heterogeneidad de la información, [Klemke, 2002] presenta la aplicación del enfoque sensible al contexto para mejorar los *procesos de mediación de información*¹⁹. La propuesta incluye un modelo conceptual y guías para la representación, evaluación y recuperación de información contextual. También en [Huang & Tao, 2004] se aplica el enfoque sensible al contexto al área de gestión de conocimiento, en este caso para el soporte en entornos de negocios. La solución propuesta también apunta a mejorar la interoperabilidad entre diferentes dominios de conocimientos, útil para actividades de negocios genéricas entre empresas físicamente distribuidas.

El enfoque sensible al contexto también ha sido aplicado en el área de la inteligencia artificial, particularmente en [Gong, 2005], donde se desarrolla un marco de modelado orientado a objetivos aplicado a la resolución de problemas basado en contexto. También se describen los mecanismos utilizados para llevar a cabo dicho procesamiento.

Por último, pero no por ello menos importante, el enfoque sensible al contexto también ha sido aplicado a diversas áreas de la Ingeniería de Software. En este entorno el foco del contexto se centra en cualquier entidad involucrada en alguna tarea del ciclo de vida de desarrollo de software mientras que la información de contexto proviene de fuentes tales como procesos, proyectos, recursos o la organización misma, entre otros. En éste área se encuentran los trabajos de [Kaltz *et al.*, 2005] donde se propone un enfoque de modelado de contexto para ser aplicado en procesos de ingeniería web destinados a la construcción de aplicaciones web sensibles al contexto. En estas aplicaciones, la navegación, la selección de contenidos y servicios y el tipo de interfaz gráfica pueden ser adaptados de acuerdo al contexto actual de los usuarios finales. Otros trabajos se enfocan en utilizar la información de contexto durante el proceso de desarrollo de software. Tal es el caso de [Araujo *et al.*, 2003] que propone utilizar información de contexto en las herramientas de soporte colaborativo utilizadas en las aplicaciones de desarrollo de software con el objetivo de mejorar la gestión del conocimiento del proceso de desarrollo en una organización, así como su aprendizaje. Los autores proponen un marco para identificar y organizar información contextual en las organizaciones de desarrollo de software así como los mecanismos para introducir dicha información en herramientas de desarrollo de software colaborativo. De la misma forma, en [David *et al.*, 2007] se presenta un modelo conceptual de contexto desarrollado para el proyecto TEAM²⁰ cuyo objetivo es desarrollar un sistema de software de código abierto, fácilmente integrable a entornos de desarrollo de software para permitir el intercambio de conocimiento descentralizado, personalizado y sensible al contexto a través de un conjunto de componentes arquitecturales. Siguiendo la misma línea, [Antunes *et al.*, 2011] propone la definición de un modelo de contexto que tiene en cuenta todas las dimensiones que caracterizan el entorno de desarrollo del desarrollador de software a nivel de proyecto. La información de este contexto, que puede ser extraída de las herramientas de gestión del proyecto –según indica la experiencia de los autores, puede ser usada para enriquecer el entorno

¹⁹Del inglés *information brokering process*.

²⁰<http://www.team-project.eu/>

de trabajo del desarrollador con conocimiento adicional para soportar su trabajo. De forma similar, en el dominio del *Desarrollo de Software Global*²¹ –en el que surgen problemas de comunicación, cooperación y coordinación debido a la distribución física de los equipos de desarrollo–, Chaves y otros [Chaves *et al.*, 2010] proponen un modelo conceptual para especificar el contexto en un entorno de Ingeniería de Software distribuido (desarrollado por los mismos autores). Esta propuesta pretende mejorar el conocimiento de los desarrolladores respecto del proceso y el contexto en el cual se generan los artefactos colaborativos utilizados en el desarrollo de software, argumentando que la especificación conceptual del contexto permite soportar el entendimiento, la resolución de problemas y la comunicación entre los agentes participantes acerca de un dominio dado. La propuesta incluye (i) la identificación de información que describe el contexto de las entidades de entorno y (ii) la definición de un modelo que facilita la comprensión de la semántica del dominio, reduciendo las brechas de comunicación. También [Henrich & Morgenroth, 2003] se enfocan en el proceso de desarrollo de software colaborativo, en este caso orientado a componentes, al describir un enfoque para buscar componentes potencialmente útiles en repositorios de la organización o redes de desarrollo de código abierto disponibles en internet. La propuesta está basada en técnicas de recuperación para documentos estructurados y utiliza información de contexto acerca del usuario y su trabajo actual para refinar las consultas. La propuesta cubre todos los tipos de artefactos creados en un proceso de desarrollo de software.

2.2.5. Discusión

Como se vio al principio de esta sección, existe un importante número de trabajos donde se proveen definiciones del término contexto a fines de ser utilizados en la aplicación del enfoque sensible al contexto en cualquier dominio. No obstante lograr un acuerdo en cuanto a una definición general del término no debiera ser el objetivo final si se pretende aplicar dicho enfoque, más bien constituye un medio para definir un modelo que permita representar la información del contexto para el propósito de la aplicación. En este sentido, la definición provista por Dey (presentada en la Sección 2.1.1) ha probado ser lo suficientemente general y concisa como punto de partida considerando la gran cantidad de trabajos relacionados que recurren a la misma. No obstante, la consideración de otras definiciones pueden dar luz a una caracterización y especificación de contexto más adecuada al propósito deseado. Por otro lado, y de igual manera, acordar sobre una definición del término sensible al contexto puede ofrecer una guía para determinar los mecanismos que serán necesarios en la aplicación del enfoque para identificar, capturar y procesar la información del contexto relevante para satisfacer el propósito establecido para el dominio de la aplicación.

En todos los trabajos citados anteriormente se provee una forma de clasificar la información de contexto según la dimensión o característica de donde proviene la misma, ya sea mediante (i) el uso de categorías predefinidas o adaptables –mediantes mecanismos propios del modelo propuesto, o (ii) requiriendo la definición de categorías propias o el reuso de categorías existentes para cada situación. Solo en algunos casos las propuestas citadas no ofrecen mecanismos de adaptación de las dimensiones de contexto. Cabe destacar al respecto que en un modelo de contexto que pretenda ser genérico no se debieran preestablecer un conjunto de dimensiones ya que estas dependerán del dominio de aplicación. Así, debieran preverse mecanismos que permitan adaptar el modelo de contexto a las necesidades particulares de dicho dominio. En este sentido también es deseable que el modelo permita la incorporación de dimensiones o fuentes de contexto existentes en el dominio de aplicación con el mínimo de esfuerzo para evitar el costo de retrabajo de traducir tales conceptos al formalismo de representación del modelo de contexto definido. De manera contraria, predefinir un conjunto de dimensiones de contexto restringe al modelo a un dominio de aplicación particular. Y si éstas dimensiones son muy genéricas la aplicación del modelo podría no ser apropiada. Estos aspectos deben ser tenidos en cuenta al momento

²¹Desarrollo de software realizado en ubicaciones geográficamente separadas de forma coordinada involucrando interacciones en tiempo real y asincrónicas.

de seleccionar una propuesta de modelado existente para aplicar el enfoque sensible al contexto en una aplicación dada.

Otro aspecto importante del modelado de información de contexto es el uso de tipos de datos. Al respecto, solo en algunos casos los trabajos citados proveen mecanismos para especificar el tipo de dato de los elementos que describen el contexto. Inclusive en los casos donde se dispone de esta información, las especificaciones disponibles carecen del detalle necesario para validar correctamente los valores que pudieran especificarse.

En cuanto al modelo de contexto, en muchos de los casos citados los autores recurren a una notación gráfica propia, en algunos casos *ad hoc*, para presentar el modelo diseñado, lo que dificulta, para estos últimos, la correcta interpretación del modelo de contexto. Este aspecto se encuentra íntimamente relacionado al formalismo utilizado para la representación de información de contexto. En esta cuestión, se observa un amplio uso del formalismo de ontologías, en armonía con lo concluido por [Strang & Linnhoff-Popien, 2004], donde puede encontrarse un análisis detallado de los formalismos existentes. El siguiente formalismo más usado es el de modelos orientados a objetos –principalmente haciendo uso de MOF/UML–, así como del formalismo de esquemas de marcas o etiquetas –tales como RDF, XML y derivados. Finalmente se observa igual adhesión entre el resto de los formalismos disponibles (modelos de clave-valor y basados en lógica). Respecto de estos últimos, cabe destacar que tanto [Kashyap & Sheth, 1997] como [David *et al.*, 2007] argumentan utilizar ontologías para definir el dominio de aplicación de contexto pero utilizan para su formalización modelos basados en lógica y orientados a objetos respectivamente, sin aprovechar de esta forma las ventajas del formalismo de ontologías. Cabe también resaltar la siguiente dificultad observada en el uso de meta-modelos para la definición de modelos de contexto, tal como se muestra en [Dos Santos, 2008]. Siguiendo dicho enfoque, un meta-modelo describe los conceptos abstractos que permiten construir los modelos que definen los conceptos de dominio que serán utilizados para describir contextos. Si se asume una interpretación del concepto de contexto como una relación existente entre entidades de algún dominio que se establece para una situación determinada –interpretación ampliamente aceptada entre los autores citados en esta sección–, el modelo de contexto debería incluir un conjunto mínimo de conceptos que describan los roles que cumplen las entidades participantes en una situación contextual (básicamente aquellas para las cuales se describe el contexto y aquellas que caracterizan el contexto de la primera). Sin embargo, en el enfoque mencionado inicialmente, se debe utilizar el meta-modelo para definir el dominio del contexto, inclusive cuando el dominio ya se encuentre definido (en el formalismo que fuera).

Adicionalmente, un problema común a los trabajos citados aquí es la falta de una base conceptual sólida cuando se trata de la cuantificación de propiedades de contexto, es decir, pocos o ningunos metadatos son especificados acerca de las escalas (y tipos de escalas) y unidades en las cuales se expresan los valores, ni los métodos utilizados para obtenerlos. Inclusive cuando esto sea hecho de forma automática por los dispositivos correspondientes, estos metadatos deben poder ser especificados durante el diseño para la selección apropiada de los dispositivos y luego para poder interpretar los resultados de forma correcta, independientemente del dispositivo utilizado para su captura. De otra forma las interpretaciones y comparaciones podrían llevar a conclusiones inválidas.

Respecto del modelo de proceso de contexto descrito en esta sección, cabe remarcar que los procesos incluidos, utilizados en la mayoría de las propuestas citadas anteriormente, ocurren durante la operación de la aplicación. No obstante, en dicho modelo se deja de lado una parte importante del proceso de contexto que permite responder a una pregunta clave de la aplicación del enfoque: ¿Cómo sabe la aplicación que debe reaccionar de tal o cual forma ante la presencia de determinadas características o propiedades del contexto? Para responder a esta pregunta, es necesario especificar en la aplicación, asociados al componente de software sensible al contexto, los valores de las propiedades de contexto ante las cuales el sistema reaccionará y cómo lo hará en cada situación o escenario. De esta forma, ante un cambio del contexto actual o solicitud de una funcionalidad sensible al contexto, la descripción del contexto asociada al componente correspondiente se utiliza como parámetro de consulta

contra la descripción del contexto actual. Estos valores pueden ser especificados durante la ejecución, como parte de la configuración o personalización de la aplicación, o en tiempo de diseño. Como ejemplo del primer caso puede considerarse una aplicación para teléfonos móviles en entornos universitarios, en la que el usuario puede configurar diferentes acciones cuando se detecte una combinación apropiada de los valores de las propiedades de contexto que indiquen que el usuario se encuentra en una clase –por ejemplo, las propiedades *cronograma de clases*, *hora del sistema* y *ubicación*. Posibles acciones son pasar el dispositivo automáticamente a silencio, redirigir todas las llamadas o activar el contestador automático. Como ejemplo del segundo caso puede considerarse una aplicación para entornos de gestión de calidad, en la que se asocia a un determinado modelo de evaluación de calidad de productos la caracterización del contexto en el cual el modelo es aplicable –por ejemplo especificando la propiedad de contexto *etapa del ciclo de vida*, ya que los atributos a medir varían según la etapa del ciclo de vida en el que se encuentre el producto [Laird & Brennan, 2006].

Finalmente, podemos ver que el enfoque sensible al contexto puede ser aplicado en cualquier área de la informática así como en cualquier dominio de aplicación siempre que exista en el mismo información relevante del contexto que afecte la forma de llevar a cabo las actividades propias del área y dominio. Requisitos casi obligatorios para tal aplicación son la definición de qué es contexto y qué es sensible al contexto en el dominio de aplicación, definir un modelo que permita representar información de contexto y su integración a la información del dominio y un modelo de proceso que describa la forma de manipular la información de contexto para llevar a cabo el comportamiento sensible al contexto deseado.

2.3. Conclusiones

A lo largo de este capítulo se han revisado las referencias consideradas relevantes para la fundamentación del objetivo de esta tesis. Concretamente, en la primera parte, se ha resaltado la importancia de los procesos de medición y evaluación en la gestión de proyectos de desarrollo de software, destacando las particularidades de la medición de calidad de artefactos de software tanto desde el punto de vista de su definición así como de su implementación. Más cercanamente relacionado al objetivo de esta tesis, se han revisado las propuestas existentes de marcos conceptuales de medición y evaluación de calidad, observando finalmente que ninguna de ellas presenta una especificación clara, concreta y sólida de los metadatos necesarios para definir las actividades en procesos de medición y evaluación, cubriendo todos los aspectos que permitan llevarlas a cabo de forma consistente y repetible. Una propuesta como tal debiera especificar de forma concisa y estructurada cada una de las propiedades que describen los detalles sobre cómo realizar la medición, evaluación e interpretación de necesidades de información acerca de los requerimientos de calidad de un producto de software (o entidades relacionadas). Adicionalmente se mostró un análisis extensivo de la importancia de contar, durante el diseño e implementación de la medición y evaluación de calidad en proyectos de software, con la información explícita y estructurada de las propiedades relevantes del contexto que afectan los resultados, interpretaciones y finalmente la toma de decisiones. Nuevamente, este aspecto se encuentra ausente en la literatura relacionada, lo que lleva al motivo de desarrollar esta tesis.

Luego, en la segunda parte del capítulo, se analiza la literatura relacionada al enfoque sensible al contexto y su aplicación a diversas áreas de dominio. Se cubren aspectos de la definición y representación de información de contexto así como del procesamiento de la misma. De los trabajos analizados se concluye que el uso de enfoques de representación basados en ontologías ofrecen una alternativa potente en cuanto al reuso y extensibilidad del modelo pudiendo ser combinados con otros enfoques de representación para facilitar la integración a los sistemas de información. No obstante dichas particularidades, las decisiones relacionadas a los aspectos mencionados deben ser tomadas considerando su aplicabilidad al propósito deseado para la aplicación, pero sin descuidar aspectos importantes del modelado de información en sistemas de información como la definición clara, explícita y sólida de los

elementos de información que componen el modelo conceptual asociado.

En conclusión, no existe en la literatura relacionada una propuesta que defina de forma clara, estructurada y explícita un marco conceptual sólido de medición y evaluación de calidad en proyectos de software que contemple todos los aspectos necesarios para la implementación coherente de programas de medición. Esto es, considerando la definición precisa de las necesidades de información sobre requerimientos de calidad, los mecanismos de medición e interpretación, todo esto soportado por una especificación explícita del contexto relevante en el que estas actividades son ejecutadas que permita alcanzar los objetivos de forma coherente.



En este capítulo se describen los fundamentos sobre los cuales se apoya el marco de medición y evaluación C-INCAMI presentado en esta tesis. Primero se describen los principios que se siguieron en la especificación del marco, que establecen un conjunto de requerimientos sobre la estructura del mismo y mecanismos asociados. Luego se describen los resultados de investigación y desarrollos sobre los cuales se construye la propuesta, tanto respecto del área de medición y evaluación como de la aplicación del enfoque sensible al contexto adoptado.

3.1. Principios

El marco de medición y evaluación propuesto en esta tesis se basa en cuatro principios fundamentales: es orientado a objetivos, basado en metadatos, centrado en la organización y sensible al contexto. Estos principios son soportados directamente por la estructura del modelo propuesto o mediante los mecanismos asociados al mismo, descriptos en la arquitectura de soporte (que será presentada en el Capítulo 7).

3.1.1. Orientado a Objetivos

El marco de medición y evaluación propuesto sigue un enfoque orientado a objetivos, es decir que responde al paradigma *top-down*, visto en el capítulo anterior, que establece que todas las tareas de medición y evaluación se enfocan en satisfacer uno o más objetivos, enunciados al comienzo del proceso.

Para que las actividades de medición y análisis sean efectivas, los objetivos deben ser definidos de forma clara y explícita [Gresse *et al.*, 1995] para lo cual se debe recurrir a una especificación estructurada de la necesidad de información y requerimientos de calidad asociados, tal como propone el enfoque GQM. La estructura a utilizar para dicho fin debe considerar que las diferentes entidades de la organización (divisiones, programas, proyectos, equipos, etc.) tienen diferentes aspectos de gestión, perspectivas e intereses. Para cada caso, las necesidades de medición se expresan en términos de los atributos de los productos, procesos y recursos involucrados [Florac *et al.*, 1997].

Para facilitar la orientación de las actividades de medición y evaluación a los objetivos definidos explícitamente el resto de los conceptos del marco deberán estar conectados de forma directa o indirecta a los objetivos de medición de forma que sea posible tenerlos presentes en todo momento para evitar desviarse de los mismos. Así, los objetivos permitirán guiar las tareas del diseño de medición y de la evaluación (cómo se llevarán a cabo las mediciones y cómo se interpretarán los valores medidos) [Florac

et al., 1997] así como la interpretación de los resultados de las mediciones hacia los objetivos originales para determinar el nivel de satisfacción de los mismos.

3.1.2. Basado en metadatos

El marco de medición y evaluación propuesto está basado en el uso de metadatos para registrar las especificaciones de cómo se llevarán a cabo la medición y la evaluación. Estas especificaciones se registran junto a los datos de los resultados correspondientes. La medición efectiva y objetiva requiere de definiciones operacionales bien definidas y claramente especificadas [Florac *et al.*, 1997]; de esta forma, los responsables de recolectar los datos de medición o de analizar los datos reportados pueden utilizar dicha especificación para llevar a cabo la recolección de datos y el análisis de los mismos de forma correcta y consistente. Los datos y metadatos definidos de esta forma permiten que los métodos utilizados para recolectar o analizar los datos (y la forma de representarlos) sean comunicados a otros y así poder determinar cómo fueron obtenidos los resultados existentes para poder ser interpretados correctamente; en consecuencia, también posibilitan la repetición de las mediciones o análisis obteniendo resultados consistentes; al mismo tiempo, es posible determinar el origen de los datos obtenidos en términos de tiempo, actividad, producto, estado, entorno, herramientas utilizadas, agente recolector, entre otros. Cuanto más objetivas sean las definiciones de los metadatos, más confiables serán los mismos para ser reutilizados y más confiables serán los resultados obtenidos.

3.1.3. Centrado en la Organización

El marco de medición y evaluación propuesto no pretende ser una mera conceptualización de los términos del dominio sino más bien una base conceptual para dar soporte tecnológico a los procesos de medición y evaluación de las organizaciones a través de la implementación de proyectos concretos. Por esto, el marco contempla la definición de los conceptos básicos necesarios para conectar aspectos administrativos relacionados a la gestión de proyectos con la especificación de los objetivos y las tareas de medición y evaluación. Más aún, el marco está diseñado para facilitar su integración a los sistemas de la organización y permitir la instauración de repositorios de metadatos y datos de medición y evaluación, útiles para establecer líneas base de hechos históricos para análisis y mejoras futuras [Florac *et al.*, 1997].

3.1.4. Sensible al contexto

El marco de medición y evaluación propuesto incluye los conceptos necesarios para especificar de forma clara y estructurada los metadatos y datos necesarios para describir el contexto relevante en el que se llevan a cabo las actividades, así como el contexto en el que las diferentes especificaciones de medición y evaluación son aplicables. Estos conceptos –y las relaciones entre ellos, componen el modelo de información requerido por el enfoque sensible al contexto (descrito en la Sección 2.2). Las descripciones de contexto son especificadas y recolectadas a través de los mismos mecanismos de medición definidos por el marco para satisfacer las necesidades de información de los requerimientos de calidad, tal como se propone en [Florac *et al.*, 1997]. En particular, el marco incluye los mecanismos necesarios para que las descripciones del contexto puedan ser capturadas directamente de los repositorios de información del dominio de la organización donde se encuentran las especificaciones de los procesos, proyectos, recursos, agentes, etc. –las entidades del entorno en el que se llevan a cabo los proyectos de medición y evaluación. El marco también define los mecanismos necesarios para utilizar la información de contexto de medición en la recomendación de aspectos de diseño y análisis de los resultados, en base a las descripciones de contexto disponibles. Este mecanismo, junto al utilizado para la captura de información de contexto, forman parte del modelo de proceso requerido por el enfoque sensible al contexto. Los detalles sobre

el enfoque sensible al contexto implementado, los conceptos y mecanismos asociados son presentados en los capítulos 4 y 5.

3.2. Enfoque de Medición y Evaluación

El marco de medición y evaluación presentado en esta tesis se construye sobre resultados de investigación previos en los que se desarrolla un marco conceptual que especifica algunos de los conceptos (y relaciones) involucrados en la medición y evaluación [Olsina *et al.*, 2005, 2008b]. A la vez, el marco presentado da soporte a resultados de investigaciones relacionadas que incluyen un proceso de medición y evaluación que describe las actividades y sus interrelaciones [Becker *et al.*, 2010] y una metodología que propone un conjunto de técnicas y herramientas para llevar a cabo dichas actividades [Olsina, 1999; Olsina & Rossi, 2002]. Estos elementos son parte de una investigación en curso [Olsina *et al.*, 2011, 2008b] para integrar una estrategia –llamada GOCAME (*Goal-Oriented, Context-Aware Measurement and Evaluation*)– para instanciar modelos de calidad para M&E. A continuación se describen brevemente estos elementos.

3.2.1. Marco INCAMI y Base Ontológica

INCAMI [Olsina *et al.*, 2005, 2008b] es un marco conceptual que define los términos (y sus relaciones) involucrados en la medición y evaluación de requerimientos no funcionales en proyectos de software y web de una organización. INCAMI está basado en una ontología [Martín & Olsina, 2003; Olsina & Martín, 2004] que define de forma explícita los conceptos, atributos y relaciones principales que se utilizan en las actividades de medición y evaluación (introducido en el Capítulo 2, Sección 1.5), y en la metodología WebQEM (*Web Quality Evaluation Method*) [Olsina, 1999; Olsina & Rossi, 2002] (introducido en el Capítulo 2, Sección 1.3). Los conceptos principales del marco INCAMI (de cuyos términos en inglés toma su nombre) son *Necesidad de Información*, *Modelo de concepto*, *Atributo*, *Métrica* e *Indicador* (mostrados en la Figura 3.1).

El marco sigue un enfoque orientado a propósitos u objetivos, ya que todas las actividades (selección de métricas y ejecución de la medición, y la definición de indicadores y ejecución de la evaluación) se enfocan en satisfacer una necesidad de información especificada inicialmente. Además, el marco está centrado en la organización, en el sentido de que la definición de los conceptos involucrados se encuentra enmarcada en el ámbito de una organización que implementa programas de medición y evaluación, contemplando aspectos de reusabilidad y consistencia de dichas actividades. Adicionalmente, INCAMI es soportado por un catálogo organizacional de medición y evaluación [Molina *et al.*, 2004] que almacena las instancias de los conceptos, definidos en el marco, para ser utilizados en las actividades de medición y evaluación, ofreciendo mecanismos de reuso que facilitan la consistencia y coherencia entre los resultados de evaluaciones de entidades para los diferentes proyectos de la organización (ver Figura 3.2).

Los conceptos del marco INCAMI están organizados en cuatro componentes principales:

- el módulo de definición de proyectos de medición y evaluación incluye los conceptos correspondientes a los diferentes tipos de proyectos utilizados como punto de partida para la planificación, asignación y gestión de recursos y actividades de medición y evaluación de forma programada (los componentes de este módulo no se muestran en la Figura 3.1).
- el módulo de definición y especificación de requerimientos no funcionales trata con la definición de la necesidad de información y el diseño de los requerimientos, mediante uno o más modelos de concepto, para instanciar, por ejemplo, en modelos de calidad interna, calidad en uso, entre otros, utilizados para guiar las actividades posteriores de medición y evaluación;

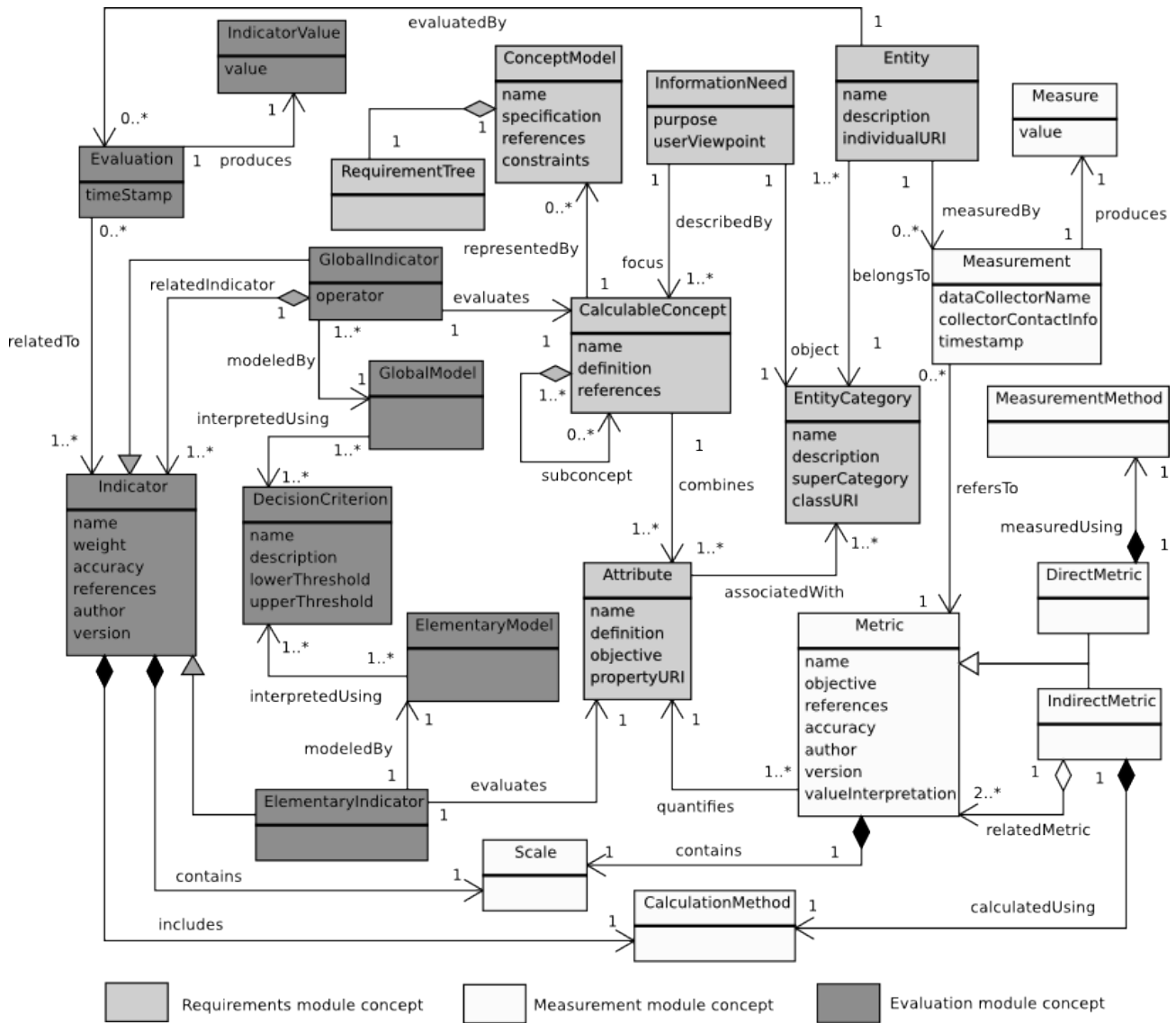


Figura 3.1: Diagrama de clases incluyendo conceptos y relaciones principales del marco INCAMI.

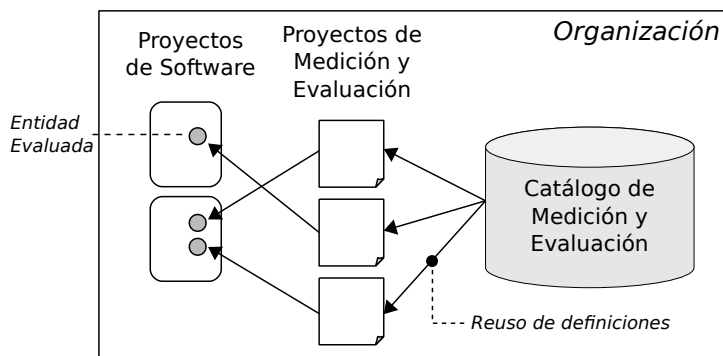


Figura 3.2: Reuso de las definiciones de conceptos de medición y evaluación en los diferentes proyectos de la organización.

- el módulo de diseño y ejecución de la medición, que trata con la especificación de las entidades concretas a ser medidas, la selección de las métricas (del catálogo organizacional) para cuantificar los atributos de los modelos de concepto seleccionados, y el registro de las mediciones obtenidas;
- el módulo de diseño y ejecución de la evaluación, que trata con la definición de los indicadores

que interpretarán los modelos de concepto seleccionados, el diseño de los criterios de decisión, y del modelo de agregación para el cálculo global que proveerá una interpretación al concepto foco de la evaluación para cada una de las entidades medidas.

3.2.2. Proceso de Medición y Evaluación

El marco propuesto en esta tesis da soporte al proceso de medición y evaluación INCAMI [Becker *et al.*, 2010] que describe la vista funcional de las actividades involucradas. Dicho proceso está basado principalmente en los estándares ISO para medición y evaluación (vistos en el Capítulo 2) e incluye las actividades e interdependencias que se muestran en la Figura 3.3. Los estereotipos `<<datastore>>` que aparecen en el diagrama representan repositorios que almacenan datos y metadatos generados y utilizados por el proceso –definiciones de atributos, conceptos, métricas, indicadores y relacionados, así como resultados de mediciones y evaluaciones (catálogos de medición y evaluación). Estos repositorios pueden ser estructurados siguiendo las definiciones del marco propuesto en esta tesis y así dar soporte al proceso asociado al marco INCAMI.

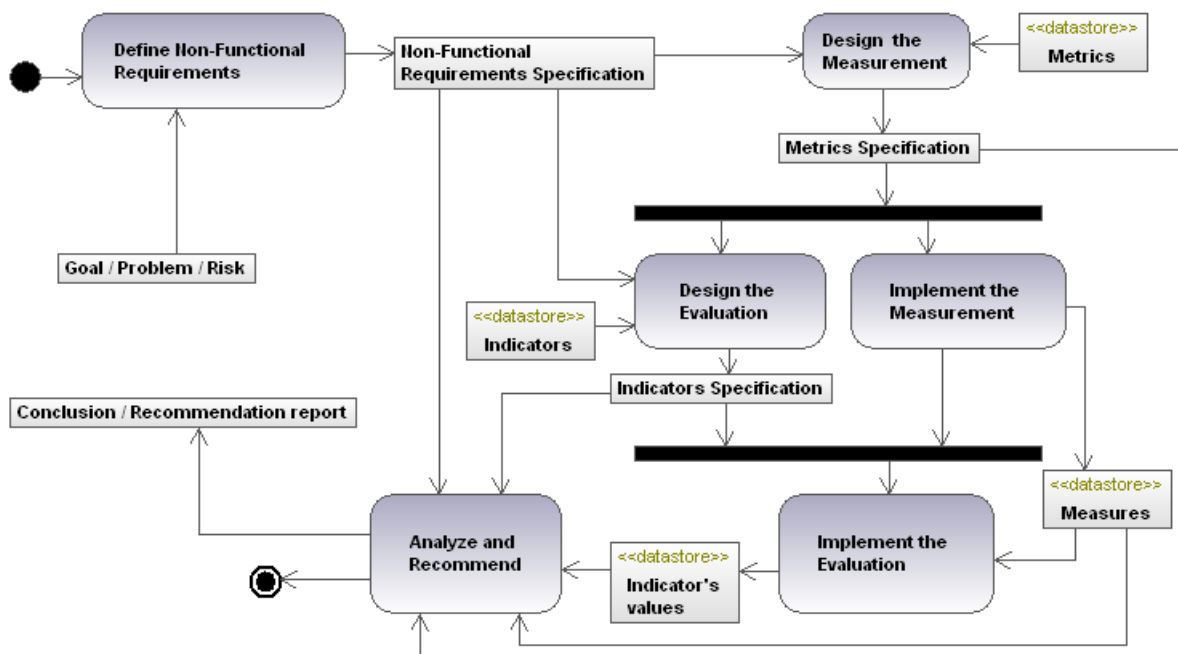


Figura 3.3: Modelo de Proceso INCAMI.

3.2.3. Metodología WebQEM

La metodología WebQEM [Olsina, 1999; Olsina & Rossi, 2002] (presentada en la Sección 2.1.3) prescribe un conjunto de métodos para ser utilizados con el proceso INCAMI, presentado en la sección anterior, para llevar a cabo la evaluación de calidad de aplicaciones web a través de cuatro fases técnicas principales. Durante la fase de *especificación de requerimientos*, se establecen los objetivos de evaluación, especificando el o los aspectos de calidad a evaluar, el propósito sobre el mismo –tal como *mejorar* o *estimar*–, las entidades a medir y el punto de vista o perfil de la evaluación –pudiendo considerar categorías como *visitante*, *desarrollador* o *administrador*, entre otras subcategorías. Luego, el aspecto de calidad a evaluar se especifica, siguiendo el estándar ISO/IEC 9126-1 [ISO/IEC 9126-1, 2001], en un modelo de calidad expresado en términos de características, las que, a su vez, se describen en términos de atributos o propiedades medibles de la entidad. El modelo de calidad representa un árbol de requerimientos que se evalúa de forma cuantitativa, desde los atributos del producto –durante

la fase de *evaluación elemental*-, hasta las características de alto nivel –durante la fase de *evaluación global*-, para llegar finalmente al grado de satisfacción de los aspectos de calidad foco la evaluación.

Durante la *evaluación elemental*, a cada atributo del modelo de calidad se le asigna una métrica que describe cómo se cuantifican los mismos (los métodos a utilizar y la forma de representación). Los resultados de las métricas permiten obtener los valores de los atributos, todos ellos heterogéneos entre sí e independientes de cualquier criterio de preferencia. Por ello, a cada métrica se asigna una función de criterio elemental que permite obtener valores de preferencia para cada atributo de forma homogénea, específicamente definidos para el objetivo de evaluación actual. Para determinar el valor de preferencia de calidad elemental (*PE*) de un atributo se puede representar una función de criterio de la siguiente forma:

$$PE = \begin{cases} 1 & \text{si } X = 0 \\ 0 & \text{si } X \geq X_{max} \\ (X_{max}-X)/X_{max} & \text{si } 0 < X < X_{max} \end{cases}$$

donde *Xmax* es algún umbral superior acordado en el dominio del atributo correspondiente. El valor de preferencia de calidad elemental resultante *PE* es interpretado como el porcentaje del requerimiento satisfecho para un atributo dado, definido en un rango entre 0 % y 100 %, normalizando de esta forma el tipo de escala y la unidad de la métrica correspondiente. Para facilitar la interpretación de preferencias se definen niveles de aceptabilidad, por ejemplo *no satisfactorio* –entre 0 % y 40 %-, *marginal* –entre 40 % y 60 %-, y *satisfactorio* –entre 60 % y 100 %. Finalmente, las métricas son aplicadas y las funciones de criterio elemental son calculadas, pudiendo utilizar las herramientas disponibles para cada caso, si las hubiere.

Durante la *evaluación global* se seleccionan un modelo de valoración y criterios de agregación asociados, como parte del diseño, para especificar las reglas de agregación de valores de preferencias elementales (atributos) y parciales (características) que permiten cuantificar el nivel de preferencia de las características de alto nivel del modelo de calidad. La metodología considera dos tipos de modelos de valoración: aditivo lineal y multicriterio no lineal, en ambos casos basados en el uso de pesos para considerar la importancia relativa de cada componente. Como ejemplo de modelo lineal aditivo puede considerarse la agregación y cálculo de la preferencia parcial/global (*PP/G*), que toma como entrada un número de valores de preferencia elementales o parciales (*PE/P_i*) y pesos relativos (*W_i*) para cada uno de ellos, de la siguiente forma:

$$PP/G = (W_1PE/P_1 + W_2PE/P_2 + \dots + W_mPE/P_m) \tag{3.1}$$

tal que, si los valores de preferencia *PE/P_i* cumplen que $0 \leq PE/P_i \leq 1$ y la suma de los pesos *W_i* cumple que $(W_1 + W_2 + \dots + W_m) = 1$; si $W_i > 0$; para $i = 1 \dots m$ entonces la ecuación 3.1 satisface $0 \leq PP/G \leq 1$. Para el caso de modelo multicriterio no lineal, la metodología propone el uso del modelo LSP [Dujmovic, 1996, 2007] (citado en la Sección 2.1.3 y explicado luego). Las preferencias parciales y globales utilizan los mismos niveles de aceptabilidad definidos para la evaluación elemental para unificar los criterios de interpretación. Finalmente, usando este modelo, se realizan los cálculos siguiendo el mecanismo de agregación paso a paso, desde las preferencias elementales hasta la preferencia de calidad global que representa el nivel de satisfacción de los requerimientos enunciados.

Al finalizar la evaluación, en la fase de *Recomendación*, todas las especificaciones de requerimientos de calidad, métricas y criterios así como los resultados de cada fase son documentados y almacenados para ser usados luego en el análisis de fortalezas y debilidades respecto de los objetivos enunciados y recomendar mejoras.

La metodología también propone el uso de una herramienta, desarrollada por los mismos autores [Olsina *et al.*, 2001], para dar soporte a proyectos de evaluación de aplicaciones web. La herramienta permite editar requerimientos no funcionales (atributos, características y sus relaciones), métricas y

funciones de preferencia (indicadores) elementales y parciales/globales, así como automatizar el cálculo correspondiente al mecanismo de agregación de preferencias para evaluar y comparar indicadores de calidad global para cada aplicación web. Los resultados son registrados en documentos enlazados apropiadamente, incluyendo información textual, tabular y gráfica a diferentes niveles de detalle, para satisfacer necesidades de información a diferentes niveles organizacionales (gerencial, de proyecto, de desarrollo).

3.2.3.1. Descripción del Modelo LSP

Una de las principales ventajas del *Modelo LSP* con respecto al modelo meramente aditivo y lineal, consiste en la posibilidad de modelar diferentes relaciones lógicas en la agregación de los parámetros incluidos en un proceso de evaluación para reflejar diferentes tipos de requerimientos [Dujmovic, 1996]. Estos tipos de requerimientos pueden clasificarse de la siguiente manera:

- *Requerimientos de Simultaneidad (o relación de conjuntividad)*: cuando los componentes incluidos en la evaluación de requerimientos son deseables simultáneamente (es decir, de forma obligatoria).
- *Requerimientos de Reemplazabilidad (o relación de disyuntividad)*: cuando los componentes incluidos en la evaluación de requerimientos son deseables de forma alternativa (es decir, no todos los componentes son obligatorios para satisfacer el requerimiento correspondiente).
- *Requerimientos de Neutralidad*: cuando dos o más preferencias de entrada son deseables de un modo independiente.

Dentro de los operadores de simultaneidad el operador *C* modela la conjunción pura, y un conjunto de operadores de *cuasiconjunción* modelan diferentes niveles de intensidad (conectores “y” flexibles): débil, medio, y fuerte, más un nivel intermedio que equivale a la media armónica (ver Figura 3.4). Los operadores de simultaneidad modelan requerimientos obligatorios de manera tal que un valor de entrada bajo no puede ser compensado por un valor alto de otra entrada para producir una preferencia de salida alta. Sin embargo, dados los diferentes niveles de intensidad de los operadores de conjunción, no todos los operadores penalizan con la misma fuerza la preferencia de salida. Concretamente, los operadores *C-* y *C--* modelan requerimientos no obligatorios, es decir, un cero en una de las entradas no producirá un cero en la salida.

Por otro lado, dentro de los operadores de reemplazabilidad, se encuentran el operador de disyunción pura *D*, así como también operadores de *cuasidisunción* para modelar relaciones de entradas en diferentes rangos de intensidad. Estos operadores modelan relaciones en las que pueden existir entradas alternativas; es decir, un bajo valor de una preferencia de entrada es bien compensado por un valor alto de alguna otra preferencia de entrada para producir una preferencia de calidad alta.

Por último, el operador de media aritmética *A* modela relaciones de neutralidad, logrando una función de criterios con una combinación perfectamente balanceada entre propiedades de disyunción y conjunción.

Los operadores vistos arriba pueden ser utilizados para combinar un conjunto de entradas de forma simple, es decir que todas las entradas son tratadas según un único operador lógico. Adicionalmente es

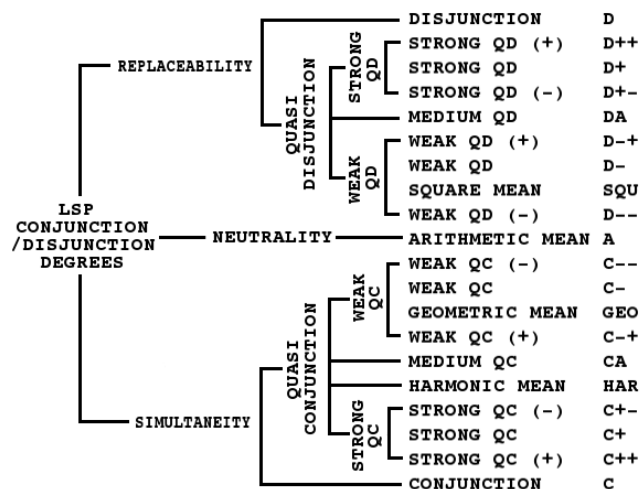


Figura 3.4: Grados de conjunción/disjunción de los operadores del modelo LSP.

posible combinar estos operadores en estructuras de agregación complejas o compuestas para modelar requerimientos más refinados, como es el caso de la absorción parcial conjuntiva que combina las entradas mediante los operadores A y CA para representar requerimientos obligatorios y opcionales en el mismo nivel, o la absorción parcial disyuntiva, que combina entradas con los operadores A y DA para representar requerimientos suficientes y opcionales.

Utilizando el modelo LSP, la agregación de las preferencias elementales, parciales y global se lleva a cabo mediante las siguientes actividades básicas:

1. Selección del tipo de relación lógica entre elementos y/o componentes (relación de simultaneidad, reemplazabilidad o neutra).
2. Selección del tipo de función conforme a la relación de entradas entre elementos y/o componentes (funciones simples o complejas¹).
3. Selección del operador conforme al nivel de intensidad de la relación lógica.
4. Selección de la importancia relativa de cada entrada conforme a los requerimientos (selección de los pesos).

Las estrategias y mecanismos para realizar estas actividades puede ser más o menos intuitivo, conforme a la experiencia de los evaluadores y el nivel de criticidad del proyecto de evaluación, pudiendo aplicar mecanismos como encuestas, estudios de análisis y sensibilidad, entre otros.

3.3. Enfoque Sensible al Contexto Adoptado

En esta sección se describe la aplicación del enfoque sensible al contexto al dominio de medición y evaluación de calidad de software. El objetivo de la misma es permitir la realización de diseños de medición más coherentes e interpretaciones más sólidas a partir de los resultados de evaluación de diferentes proyectos de la organización. En este entorno, las descripciones de contexto serán especificadas tanto para describir la situación real de proyectos de medición y evaluación como para describir el contexto donde ciertos elementos de información reusables –por ejemplo métricas, modelos de calidad, etc.–, pueden ser aplicados coherentemente. Estas descripciones puede ser utilizadas luego –mediante mecanismos de comparación–, como base para un sistema de recomendación en proyectos de medición y evaluación.

Cabe aclarar que lo que se propone aquí no es un marco de modelado de contexto independiente del dominio, sino una aplicación del enfoque sensible al contexto para el dominio mencionado. No obstante, se debe aclarar que el enfoque desarrollado aquí y la arquitectura de soporte asociada pueden ser utilizados en otros dominios de la Ingeniería de Software, inclusive en otras áreas, siempre y cuando se adapte a las particularidades de los mismos.

Como se introdujo en el Capítulo 1 y se describió en el Capítulo 2, la aplicación del enfoque sensible al contexto requiere decidir los siguientes elementos:

1. una definición o caracterización de qué es contexto en el entorno del dominio de aplicación (su alcance);
2. un modelo de información para representar la información de contexto relevante y la integración y contextualización de la información del dominio respectivo;
3. un modelo de proceso que describa los mecanismos necesarios para capturar, procesar y utilizar la información de contexto en el dominio de aplicación.

¹En esta tesis solo se utilizán funciones de agregación simple para modelar diferentes tipos de requerimientos.

A continuación se enuncian los requerimientos identificados para la aplicación del enfoque sensible al contexto al marco de medición y evaluación propuesto en esta tesis. Luego, en base a dichos requerimientos, se describen las decisiones para cada uno de los elementos, dejando los detalles de aplicación al dominio correspondiente para los Capítulos 4 y 5, donde se describe la propuesta que motivó esta tesis.

3.3.1. Requerimientos

Como se indicó en los Capítulos 1 y 2 la decisión sobre los elementos que constituyen la aplicación del enfoque sensible al contexto, enumerados anteriormente, debe ser tomada a la luz de las necesidades o requerimientos particulares de la aplicación del mismo. Estos requerimientos ayudan a responder las preguntas sobre qué alternativa elegir para cada uno de los elementos del enfoque. A continuación se enumeran y describen los requerimientos deseables de la aplicación del enfoque sensible al contexto a la gestión de proyectos de medición y evaluación de calidad de software, objetivo principal de esta investigación:

- R1** El modelo de información debe reflejar la caracterización de contexto (que se verá en la siguiente subsección) tal como se entiende para el dominio de M&E.
- R2** El modelo de información de contexto debe ser independiente del dominio al que pertenece el objeto sometido a las actividades de M&E y poder ser adaptado a cualquier dominio dado. El fundamento de este requerimiento se debe a que en cualquier caso, el contexto no es un dominio más sino que proviene de un dominio existente.
- R3** La información de contexto debe poder ser validada contra un modelo de dominio tanto a nivel de estructura como de instancia.
- R4** El modelo de contexto debe permitir la representación de información de contexto de forma integrada a la información manipulada por los sistemas existentes en el dominio de aplicación.
- R5** La información de contexto debe ser consistente, es decir, evitar la ambigüedad en su descripción.
- R6** El modelo debe ser lo suficientemente simple para ser especificado y procesado manteniendo un balance entre costo y desempeño en el sistema.
- R7** La información de contexto debe poder ser especificada (i) para cada proyecto de medición y evaluación, como parte de la especificación de requerimientos del mismo y (ii) de forma anticipada e independientemente de cualquier proyecto de M&E, para cada elemento de información reusable en las actividades de diseño.
- R8** Las descripciones de contexto deben poder ser comparadas tanto para determinar la similitud entre los contextos de dos proyectos de M&E dados, como para determinar la aplicabilidad de un elemento de información reusable en un proyecto de M&E dado.

3.3.2. Caracterización del Contexto

A partir de las definiciones de contexto encontradas en la literatura (vistas en Sección 2.1) y un correspondiente análisis de las mismas, nos lleva a identificar un conjunto de elementos en común que permiten caracterizar a la información de contexto (ver Figura 3.5):

- Es relativa a un objeto o entidad de interés específico;
- Es aquella relevante respecto de:

- la tarea específica relacionada a esa entidad,
 - la intención específica, relacionada a esa tarea, para con la entidad,
 - la interacción de la entidad de interés con otras entidades del contexto, respecto de la tarea y propósito;
- Está compuesta o caracterizada por un conjunto de propiedades relevantes, cada una con una conceptualización asociada (es decir, definición y estructura) que hace explícita su interpretación. Estas propiedades pueden pertenecer a la entidad de interés o entidades relevantes relacionadas a la primera.

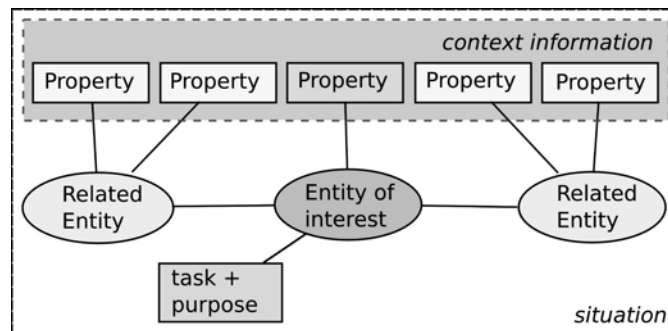


Figura 3.5: Caracterización de la información de contexto.

Resumiendo la caracterización anterior, para el propósito de esta tesis se considera información de contexto, respecto de una entidad de interés, al conjunto de propiedades relevantes (y sus valores) de las entidades involucradas en una situación dada. Por *situación* se entiende la suma de la entidad de interés, la tarea e intención para con la misma, y la interacción entre ésta y otras entidades relevantes; y por *relevante* se entiende que puede afectar la forma en que la información es interpretada respecto de la situación en cuestión.

Aunque la caracterización anterior ha sido elaborada con la intención de ser utilizada en el dominio de aplicación de medición y evaluación de calidad de software, ha sido pensada de forma independiente de cualquier dominio y, por lo tanto, creemos que puede ser aplicable a cualquiera de ellos. Por ejemplo, si consideramos el caso de la computación ubicua, el usuario móvil constituye la entidad de interés para la cual se describe el contexto, la situación se describe mediante la tarea e intención del usuario –asociados al dominio de la aplicación como por ejemplo una aplicación de guía turística–, y las entidades asociadas al mismo –por ejemplo, los lugares o dispositivos en la vecindad del usuario–; los aspectos relevantes del contexto son aquellos relacionados al tiempo, ubicación, dispositivo, actividad, rol, etc.

3.3.3. Modelado y Representación de Contexto

A partir de los requerimientos enunciados, se debe decidir sobre un enfoque de modelado para representar la información de contexto tanto a nivel de diseño como de implementación, considerando que dicha información deberá ser manipulada por un sistema de software y almacenada en repositorios apropiados. Para la elección del enfoque de modelado se consideraron un número de alternativas [Strang & Linnhoff-Popien, 2004], analizando los *pros* y los *contra* de cada una de ellas para el dominio de aplicación de esta tesis. Finalmente se decidió sobre un enfoque mixto para cubrir diferentes aspectos en la representación de contexto.

Por un lado, se utiliza el enfoque de modelado *orientado a objetos* para describir el diseño conceptual de los términos y relaciones utilizados para representar la información de contexto, debido a su expresividad y, al mismo tiempo, su facilidad de entendimiento. Particularmente se utilizó UML (*Unified*

Modeling Language)² debido a su robustez como formalismo de modelado, su nivel de estandarización y su amplio soporte. Este modelo es utilizado luego como base para la implementación de un sistema de software que de soporte tecnológico al mismo. Utilizando este formalismo y considerando que en la caracterización vista anteriormente las propiedades de contexto no son más que los atributos medibles de alguna entidad, el modelo de contexto se construye sobre una base conceptual de metrología –constituida por los conceptos y relaciones del área de medición definida en [Olsina & Martín, 2004] (requerimiento R1). De esta forma, las propiedades de contexto pueden ser especificadas y cuantificadas utilizando los conceptos, relaciones y mecanismos definidos en el marco conceptual INCAMI [Olsina *et al.*, 2005, 2008b], basado en la ontología citada. De esta forma, el modelo de contexto se beneficia de la consistencia y robustez de contar con metadatos asociados, facilitando el reuso y comparabilidad de los datos correspondientes (requerimiento R8). En el modelo resultante, representado en la parte derecha del diagrama de alto nivel de la Figura 3.6 –el modelo de contexto detallado se presenta en el Capítulo 4– el contexto se estructura como un conjunto de definiciones de propiedades de contexto y de las entidades a las cuales pertenecen (según corresponda, dependiendo de la categoría de la entidad). Los valores de las propiedades de contexto se representan como mediciones realizadas a la entidad correspondiente, obtenidas como resultado de aplicar una métrica que cuantifica dicha propiedad. Cabe aclarar que cada propiedad de contexto puede asumir más de un valor en la descripción de un contexto particular (estos detalles son profundizados en el Capítulo 5). El conjunto de todos los valores medidos para cada propiedad de contexto constituyen la descripción de un contexto concreto. De esta forma, para cada contexto se dispone, no sólo de los valores (datos) que lo describen sino también de las definiciones (metadatos) de las propiedades que lo caracterizan, de las entidades involucradas en el contexto a las cuales pertenecen –y su categoría– y de las métricas a partir de las cuales se obtienen los datos (el método utilizado y la escala en la que se representan los valores). El uso de metadatos para definir los elementos que componen una descripción de contexto permiten especificar información de contexto de forma consistente (requerimiento R5). Además, dado que el modelo es independiente del dominio y se construye sobre los conceptos y relaciones de medición de un marco existente, los detalles de cómo se lleva a cabo la cuantificación de las propiedades quedan ocultos al modelo de contexto, manteniendo la simplicidad de éste último tanto para la especificación como para el procesamiento de la información de contexto (requerimiento R6).

Por otro lado, ya que las propiedades y entidades utilizadas para describir el contexto pertenecen al dominio de aplicación descripto, se utiliza el enfoque de modelado mediante el formalismo de *ontologías* para representar las definiciones de los conceptos y propiedades (metadatos) del dominio de aplicación, a partir de las cuales se especificarán las propiedades de contexto y las categorías de las entidades a las que pertenecen –como se muestra en la parte izquierda de la Figura 3.6. De esta forma, las propiedades y entidades utilizadas para describir contexto mantienen referencia a los conceptos y propiedades definidas en el dominio de aplicación (requerimiento R2). Cabe aclarar que no es obligatorio que las propiedades de contexto se especifiquen exclusivamente a partir de definiciones de atributos del dominio de aplicación –ya que estos podrían no haber sido especificados hasta el momento– si no que pueden ser definidas de forma independiente para tal fin. No obstante, y por razones de consistencia, las propiedades (y categorías de entidad) del dominio especificadas para describir contextos³ debieran ser especificadas también en el espacio de información del dominio de aplicación.

El uso del formalismo de ontologías permite acceder a sus capacidades de intercambio y reuso de conocimiento, inferencia lógica y la posibilidad de representar la información de dominio en estructuras de datos utilizables por computadoras (requerimiento R4). Por lo anterior, el modelo de contexto puede ser conectado y reusado en aplicaciones existentes ya que los conceptos del modelo de contexto no especifican qué situación (entidad) caracterizan, cuáles son las entidades sensibles al contexto o qué

²<http://www.uml.org>

³Otras propiedades independientes del dominio podrían ser especificadas, por ejemplo, asociadas del proceso de M&E mismo.

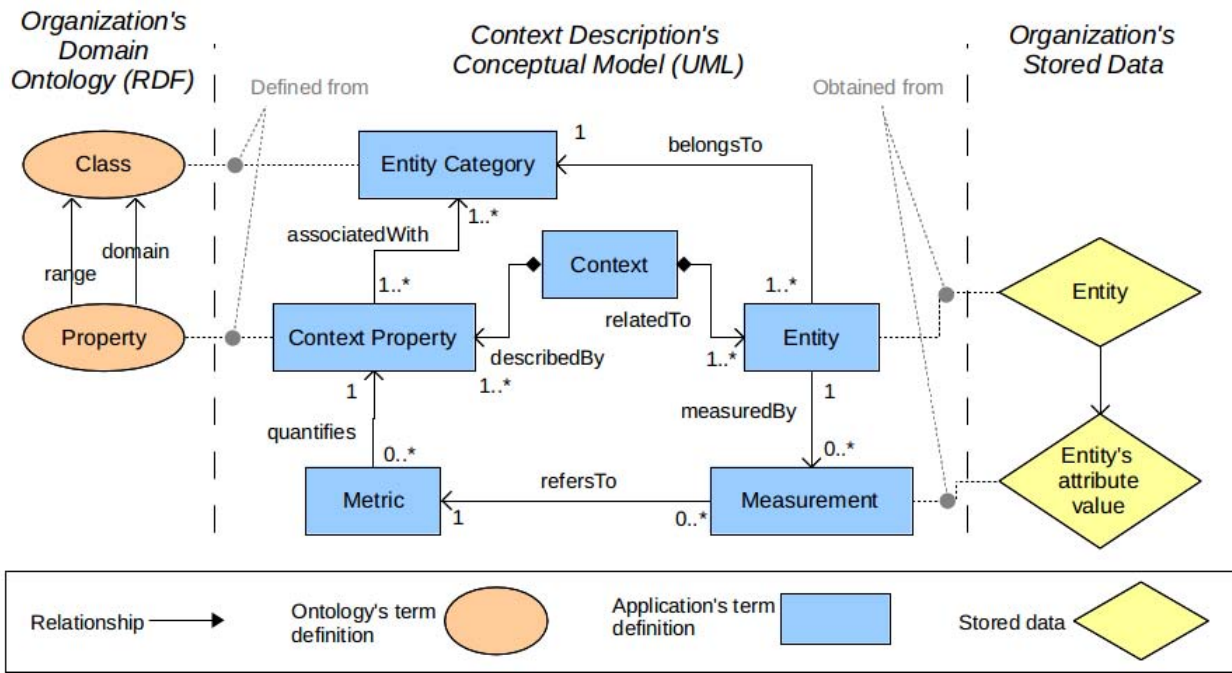


Figura 3.6: Estructura de alto nivel del modelo de contexto y dependencias con definiciones y datos del dominio de aplicación.

propiedades utilizar en la descripción de contextos, más bien el dominio que implementa el enfoque es responsable de decidir qué conceptos serán contextualizados estableciendo las relaciones con los elementos correspondientes del modelo de contexto y qué propiedades del dominio serán utilizadas para describir un contexto. Por lo tanto, éste modelo podría, en principio, ser usado para describir información de contexto para cualquier dominio, aunque fue concebido para ser aplicado a un entorno de Ingeniería de Software (requerimiento R7).

Para la implementación del formalismo de ontologías se utilizó el lenguaje RDF/S (*RDF Schema*) [W3C-RDFS, 2004] debido a que, por un lado, los repositorios incluidos en el soporte tecnológico existente para el marco INCAMI, sobre el cual se construye el contexto y en el que se integra el enfoque, están estructurados en dicho lenguaje, y por otro lado, debido al estado de estándar del lenguaje y a la disponibilidad de un conjunto importante de herramientas de soporte para su almacenamiento, gestión y validación. De esta forma, se pueden implementar *catálogos de dominio* que almacenen metadatos y datos del dominio de aplicación así como *catálogos de contexto* que contengan especificaciones de las propiedades y entidades utilizadas para describir contextos para dicho dominio de aplicación de la misma forma que se mantienen catálogos de M&E –como se vio en el punto 3.2.1 (los aspectos de la arquitectura de soporte serán presentados en el Capítulo 7). Así, también es posible obtener de dichos catálogos los valores correspondientes a las instancias de las entidades y sus propiedades involucradas en la descripción de contextos usando el modelo de objetos descrito anteriormente. En consecuencia, tanto las definiciones de las propiedades y entidades del contexto como sus valores son validables respecto de los datos del dominio de la organización (requerimiento R3). Por supuesto, la decisión de utilizar RDF impone una restricción en la implementación del enfoque en el sentido que el dominio de aplicación debe estar codificado en el mismo formalismo. Sin embargo, creemos que actualmente RDF se ha convertido en un estándar industrial para el intercambio de información. En el peor de los casos, existe un conjunto de herramientas que permiten traducir esquemas de bases de datos relacional a RDF, que pueden utilizarse para cubrir la brecha (ver desarrollos del “W3C Workshop on RDF Access to Relational Databases” [W3C-RDB2RDF-Workshop, 2007] y del “RDB2RDF Working Group” [W3C-RDB2RDF-Group, 2011]).

3.3.4. Procesamiento de Información de Contexto

El último de los elementos a definir para la aplicación del enfoque sensible al contexto es un modelo de proceso, es decir, los mecanismos mediante los cuales la información de contexto será especificada, capturada y procesada (tal como se describió en la Subsección 2.2.3) para ser utilizada en el dominio de aplicación con un propósito particular.

Respecto de los mecanismos para especificar y capturar información de contexto se decidió por un enfoque de contexto pasivo, es decir que el contexto es especificado y descrito en algún momento previo a su utilización y es asociado al elemento de información de dominio correspondiente. Esta descripción de contexto será utilizada posteriormente como parámetro de consulta para la selección de la información de dominio apropiada según la tarea. La elección de este enfoque se debe a que en proyectos de M&E en entornos de Ingeniería de Software el contexto es relativamente estable considerando un período de tiempo determinado, particularmente aquel durante el cual se lleva a cabo la medición y evaluación. Cabe agregar que para la descripción del contexto primero se seleccionan las propiedades de la entidades relacionadas a la entidad de interés que se consideran relevantes y luego se cuantifican utilizando la especificación de la métrica asociada pudiendo acceder a los valores correspondientes almacenados en los repositorios de información de dominio de la organización (según se comentó en la subsección anterior).

Respecto del procesamiento y uso de la información de contexto se identifican dos casos para los cuales se diseñan sendos mecanismos de comparación de contexto (requerimiento R8). Por un lado se desea determinar el grado en que una entidad o elemento de información particular es aplicable a una situación o entorno real dado, para lo cual se comparan las descripciones del contexto de la situación dada con la descripción del contexto ideal en el cual la entidad es aplicable. Por otro lado se desea determinar el grado de similitud de dos situaciones reales dadas para lo cual se comparan las descripciones de contexto correspondientes a cada una. En ambos casos, el mecanismo debe producir un valor que exprese de forma objetiva el grado de similitud o aplicabilidad (según corresponda) de los contextos comparados. La estructura de los datos necesaria para llevar a cabo esta comparación es presentada en el Capítulo 4, mientras los mecanismos de comparación son detallados en el Capítulo 5.

Parte II

MEDICIÓN Y EVALUACIÓN ORIENTADA A OBJETIVOS Y SENSIBLE AL CONTEXTO

El Marco Conceptual C-INCAMI



En este capítulo se describen y definen los términos y relaciones que constituyen el marco conceptual C-INCAMI, objeto central de esta tesis. Además se describe el mecanismo por el cual el marco de M&E puede ser integrado al dominio de aplicación de la organización que lo implemente.

4.1. Introducción al Marco C-INCAMI

El marco conceptual C-INCAMI provee un modelo de dominio que define todos los conceptos y relaciones necesarias para implementar un proceso de M&E en entornos de Ingeniería de Software de una organización, siguiendo los principios descritos en el Capítulo 3, Sección 1 –orientado a objetivos, basado en metadatos, centrado en la organización y sensible al contexto. Adicionalmente, el marco fue diseñado atendiendo a las características deseables en este tipo de conceptualizaciones (también identificadas en el Capítulo 1) cuya carencia en las propuestas existentes motivaron este trabajo.

El nombre del marco propuesto responde al término *Contextual INCAMI*, para destacar el marco sobre el cual fue construido (según se indicó en los capítulos 1 y 3) y el enfoque sensible al contexto aplicado al mismo (descrito en el Capítulo 3, Sección 3), que constituye una de las principales contribuciones de esta tesis. Entonces, C-INCAMI se basa en los conceptos centrales de *contexto*, *necesidad de información*, *modelo de concepto*, *atributo*, *métrica* e *indicador*. Respecto de esta contribución, cabe destacar que el agregado de los metadatos apropiados permite describir de forma consistente la información del contexto que caracteriza la entidad bajo análisis, relevante a una necesidad de información particular. Estos metadatos permiten la captura e interpretación consistente y coherente de las descripciones (datos) de contexto, más aún si se encuentra almacenada en catálogos reusables de la organización. El modelo definido en C-INCAMI permite también contextualizar elementos de información reusables de medición y evaluación (tales como modelos de conceptos, métricas, entre otros) según se requiera, proveyendo las bases para un mecanismo de recomendación basado en comparaciones de similitud. El objetivo final de incorporar información de contexto es lograr una mejor interpretación y toma de decisiones a partir de los resultados de medición y evaluación de calidad en entornos de Ingeniería de Software.

Al igual que su antecesor, el marco C-INCAMI puede ser utilizado para especificar y estructurar los datos y metadatos generados y usados durante las actividades involucradas en el proceso de M&E, instanciando los conceptos de forma apropiada. También, siguiendo estas especificaciones, se pueden definir catálogos organizacionales para almacenar información reusable entre los diferentes proyectos de la organización. Así, los metadatos y catálogos de M&E pueden ser usados para construir herramientas de soporte para automatizar todo o parte del proceso. Estos aspectos, que conforman parte de la contribución de esta tesis, son cubiertos en el Capítulo 7.

El marco C-INCAMI está estructurado en módulos o paquetes, a saber (ver Figura 4.1):

módulo *c-incami*: es el módulo principal del marco y contiene los conceptos relacionados a la gestión de proyectos de medición y evaluación y al resto de los módulos del marco;

módulo *requirements*: contiene los conceptos y relaciones asociados a la definición de los objetivos de medición (siguiendo el enfoque GQM) y la especificación de requerimientos no funcionales;

módulo *context*: contiene los conceptos y relaciones asociados a la especificación y descripción de información de contexto;

módulo *measurement*: contiene los conceptos y relaciones asociados a la definición de las actividades de diseño e implementación de la medición;

módulo *evaluation*: contiene los conceptos y relaciones asociados a la definición de las actividades de diseño e implementación de la evaluación;

Vale mencionar que, aunque el contexto es parte de la especificación de requerimientos no funcionales, y por lo tanto, utilizado en todas las actividades de diseño del proceso así como en el análisis final y recomendación, se mantiene como un módulo separado por razones de flexibilidad e independencia. Por lo tanto, la información de contexto puede ser utilizada también para contextualizar objetos de ambos de los módulos de medición y evaluación.

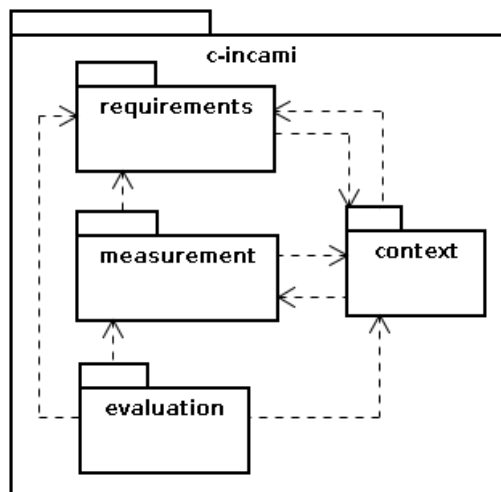


Figura 4.1: Organización de paquetes o módulos (UML) del marco C-INCAMI.

Como puede observarse en la figura mencionada, todos estos módulos se encuentran interrelacionados. El módulo de requerimientos utiliza los conceptos del módulo de contexto para describir la información de contexto real de un proyecto, mientras los módulos de medición y evaluación lo utilizan para describir el contexto en el cual son aplicables los diferentes elementos de información reusables que se definen durante tales actividades. Los módulos de medición y evaluación también dependen del módulo de requerimientos ya que cada definición de estas actividades apuntan a satisfacer una necesidad de información correspondiente a un proyecto particular. De forma similar, el módulo de evaluación depende de las especificaciones y valores registrados mediante el módulo de medición ya que son éstas las que serán utilizadas como entrada para su interpretación. Finalmente, el módulo de contexto depende de los módulos de requerimientos y medición ya que contienen los conceptos sobre los cuales se construye el modelo de información de contexto (tal como se describió en el Capítulo 3, Sección 3).

A continuación se describen de forma detallada los módulos del marco, incluyendo, luego de cada descripción las definiciones de los términos correspondientes.

4.2. Gestión de Proyectos

El módulo principal del marco C-INCAMI define los metadatos básicos para describir proyectos de medición y evaluación de calidad de entidades en entornos de Ingeniería de Software y web. Tal como se muestra en la Figura 4.2, el módulo incluye un concepto abstracto *Project* que describe de forma genérica la información que todo proyecto debe incluir, tal como su nombre, una descripción del mismo, el nombre del responsable e información de contacto, fecha de inicio y fin. A partir de este concepto abstracto se definen el resto de los proyectos concretos del marco.

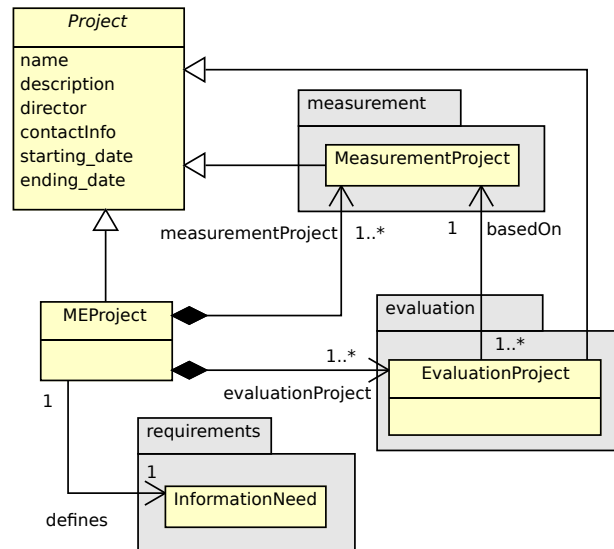


Figura 4.2: Diagrama de clases (UML) describiendo los conceptos y relaciones del módulo principal de C-INCAMI.

El concepto principal de este módulo es el de *Proyecto de Medición y Evaluación (MEProject)* para el que se especifica una *Necesidad de Información* que establece los objetivos y los requerimientos no funcionales que guían el resto de las actividades (como se verá en la sección siguiente). Este concepto pretende ser utilizado para organizar los conceptos del marco involucrados en el diseño e implementación de la medición y evaluación, y como punto de referencia para la planificación de las tareas –cronograma, asignación de recursos, roles, etc. Cada *Proyecto de M&E* se divide en subproyectos de medición y de evaluación –relacionados entre sí e incluidos en los módulos correspondientes– considerando que cada uno de estos últimos pueden ser administrados de forma separada, permitiendo así la planificación coordinada de los mismos. Cada *Proyecto de M&E* puede contener uno o más *Proyectos de Medición (MeasurementProject)*, todos ellos satisfaciendo la necesidad de información común definida en el primero, pudiendo llevarlos a cabo en paralelo o en diferentes momentos. A su vez, a partir de cada *Proyecto de Medición* pueden llevarse a cabo uno o más *Proyectos de Evaluación (EvaluationProject)*, satisfaciendo la misma necesidad de información, pudiendo reflejar cada uno de ellos diferentes criterios de evaluación. Esta separación en subproyectos facilita la trazabilidad y consistencia para realizar análisis intra e interproyectos.

Respecto del enfoque sensible al contexto seguido, cabe destacar que cada *Proyecto de M&E* es la unidad mínima a la cual se asocia la información –datos y metadatos– que describe el contexto que caracteriza a la entidad bajo análisis, como parte de la especificación de requerimientos no funcionales. Esto provee la base para una mejor interpretación y comparación de los resultados de proyectos de M&E.

4.3. Especificación de Requerimientos No Funcionales

En este módulo se definen los metadatos de los conceptos que permiten especificar los objetivos de medición y evaluación incluyendo la especificación de los requerimientos no funcionales. En la Figura 4.3 se muestra un diagrama ilustrando los conceptos y relaciones del módulo. Las definiciones de estos conceptos, según se especifican en la ontología correspondiente [Martín & Olsina, 2003; Olsina & Martín, 2004], son presentadas a lo largo de la sección.

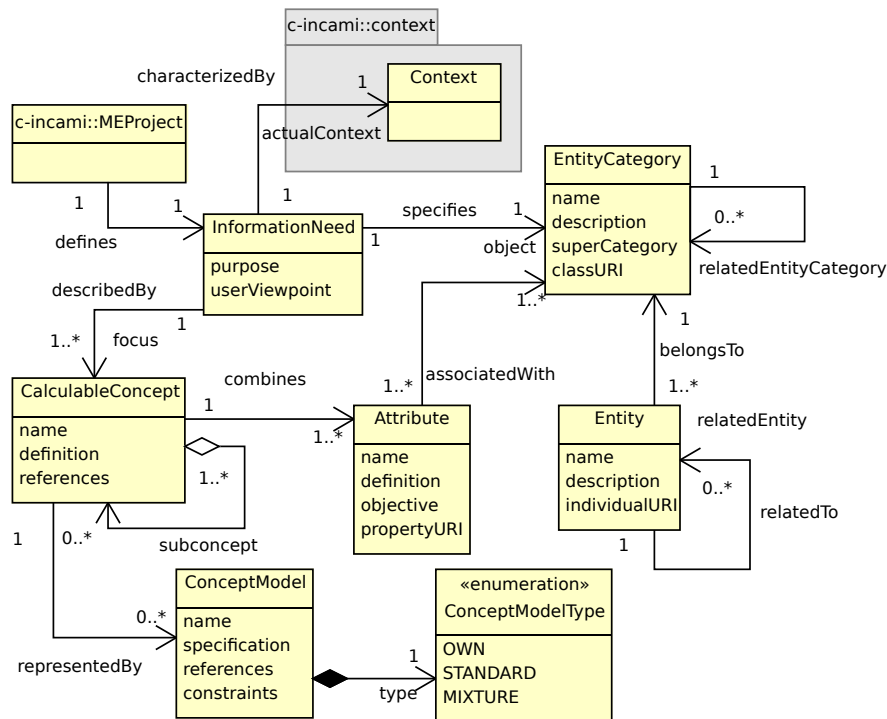


Figura 4.3: Diagrama de clases (UML) mostrando los conceptos y relaciones del módulo *requirements* del marco C-INCAMI.

El concepto principal de este módulo es *Necesidad de Información (InformationNeed)*, definida para un Proyecto de M&E, que especifica, siguiendo el enfoque GQM, la siguiente información:

- el propósito de llevar a cabo la evaluación, por ejemplo, “comprender”, “predecir”, “mejorar”, “controlar”, etc.;
- el concepto foco a ser evaluado –por ejemplo “calidad”, “calidad en uso”, etc.;
- la categoría de la entidad que será evaluada (*EntityCategory*) –por ejemplo “*Módulo de software*”, “*Servidor de Aplicaciones*”, “*Procedimiento de Prueba*”, etc.- y las categorías de alto nivel (el atributo *superCategory*) de interés para la Ingeniería de Software y Web, que comprenden *recursos, procesos, productos, servicios, proyectos*, entre otros;
- el punto de vista del usuario, es decir del interesado en los resultados de la evaluación, refiriéndose al rol del mismo, por ejemplo “*desarrollador*”, “*usuario final*”, “*gerente*”, etc.
- el contexto (*Context*) que caracteriza la situación, definida por los puntos anteriores, para un Proyecto de M&E particular.

Cabe aclarar que, mientras la necesidad de información especifica el tipo o categoría de la entidad a evaluar, las mediciones y evaluaciones se realizan sobre entidades concretas (*Entity*) correspondientes a esa categoría, que son especificadas luego en un *Proyecto de Medición*. Se debe tener en cuenta también que, según se observa en la Figura 4.3, las entidades pueden tener entidades relacionadas cuyos atributos pueden ser necesarios para cuantificar de forma indirecta los atributos de la entidad objeto de la evaluación. De forma análoga, es posible especificar las mismas relaciones entre categorías de entidad correspondientes.

DEFINICIÓN 1 (InformationNeed) Información necesaria para gestionar objetivos, metas, riesgos y problemas.

DEFINICIÓN 2 (EntityCategory) (*sinon.* objeto) Categoría a la cual pertenecen las entidades que son caracterizadas midiendo sus atributos.

DEFINICIÓN 3 (Entity) Un objeto concreto que pertenece a una categoría de entidad.

El concepto foco especificado en la necesidad de información corresponde a un Concepto Calculable (*CalculableConcept*) que representa una característica abstracta de calidad deseable para alguna entidad. En este caso, el concepto foco constituye la característica de más alto nivel de los requerimientos no funcionales a evaluar sobre la entidad correspondiente. Además de su *nombre* y *definición*, el modelo permite registrar las *referencias* o fuentes de donde se obtuvo el concepto (por ejemplo “ISO 9126-1”).

Para permitir la medición y evaluación cuantitativa de los requerimientos no funcionales de una entidad respecto de un determinado concepto, el marco permite especificar un Modelo de Concepto (*ConceptModel*) que relaciona de forma jerárquica conceptos calculables en múltiples niveles intermedios (de subconceptos) –en la forma de un árbol de requerimientos, al igual que los modelos de calidad de software definidos por los estándares [ISO/IEC 9126-1, 2001] e [ISO/IEC 25010, 2011]-, combinando en el último nivel Atributos (*Attribute*) cuantificables o medibles sobre la entidad. Opcionalmente, también es posible asignar atributos a los conceptos de nivel intermedio o alto, aunque si existieran más de uno podrían agruparse en un nuevo concepto de nivel intermedio. Por otro lado, si se asociara únicamente un atributo a un determinado concepto de bajo nivel, dicho concepto podría ser obviado, agregando el atributo directamente al concepto del nivel superior inmediato. El marco permite registrar para un modelo de concepto su *especificación*, que contiene una representación formal o semiformal del modelo, un conjunto de *restricciones* aplicables a la forma de composición del modelo y las *referencias* a las fuentes de donde pudo haber sido obtenida la especificación del modelo. Para la construcción de un Modelo de Concepto, pueden utilizarse modelos predefinidos en estándares, propios o adaptar modelos existentes, especificado por el tipo de modelo (*ConceptModelType*). Los modelos de conceptos constituyen la columna vertebral del proceso, y del marco mismo, ya que permiten guiar las actividades posteriores de medición y evaluación.

Un atributo representa una propiedad observable o medible asociada con (relación *associatedWith*) una determinada categoría de entidad, en particular, a aquella que constituye el objeto de la evaluación. Para cada atributo se conoce, además del *nombre* que lo identifica, una *definición* del mismo y el *objetivo* por el cual es utilizado en un proyecto de M&E. Además, con miras al diseño de modelos de concepto, cada atributo se encuentra asociado de forma preestablecida a un determinado concepto calculable (relación *combines*) para promover la coherencia y consistencia de los modelos diseñados entre los diferentes proyectos de la organización. En este sentido, diferentes organizaciones pueden asociar un determinado atributo a uno u otro concepto dependiendo de su interpretación, aunque en muchos casos ciertas relaciones son más evidentes. Tal es el caso del atributo *Invalid Links*, claramente asociado al concepto *Reliability>Availability* [ISO/IEC 25010, 2011]. Por otro lado, la asociación de un atributo con diferentes conceptos podría ser válidamente argumentable, tal como el caso del atributo *How-to-buy help availability* que puede ser asociada tanto al concepto *Usability>Learnability* [ISO/IEC 25010, 2011] como al concepto *Information Quality>Suitability>Coverage* [Olsina et al., 2009]. De la

misma forma, el atributo *Readability of Text* podría ser asociado tanto al concepto *Learnability* como al concepto *Information Quality*.

DEFINICIÓN 4 (CalculableConcept) (*sinon.* característica, factor, concepto medible) Relación abstracta entre atributos de categorías de entidad y necesidades de información

DEFINICIÓN 5 (ConceptModel) El conjunto de subconceptos y las relaciones entre ellos que proveen la base para especificar los requerimientos no funcionales para un concepto y su posterior evaluación o estimación.

DEFINICIÓN 6 (Attribute) (*sinon.* propiedad) Una propiedad abstracta o física asociada a una categoría de entidad, medible sobre una entidad concreta perteneciente a dicha categoría.

Cabe destacar que la posibilidad de disponer de los metadatos que describen cada uno de los conceptos mencionados permite especificar de forma clara y explícita la necesidad de información de medición y evaluación así como los requerimientos no funcionales asociados. Inclusive, la especificación de las categorías de entidad, las entidades mismas y sus atributos pueden ser obtenidas del espacio de información de la organización almacenado en repositorios y memorias de proyectos –siempre y cuando se encuentren disponibles– tal como se describirá luego en la Sección 4.6. Las especificaciones de todos estos conceptos pueden luego ser almacenadas en catálogos de M&E reusables para todos los proyectos de la organización, para facilitar la coherencia y consistencia entre los mismos.

4.3.1. Especificación de Información de Contexto

El módulo *context* (ver Figura 4.4) incluye la definición de los metadatos (conceptos, atributos y relaciones) necesarios para describir contextos y contextualizar –es decir, asociar descripciones de contexto a– elementos de información de dominios específicos, así como los conceptos necesarios para llevar a cabo la comparación de similitud de descripciones de contexto [Molina & Olsina, 2007b,a, 2008]. El modelo de contexto fue diseñado como un módulo separado del resto de los módulos del marco C-INCAMI por cuestiones de simplicidad y flexibilidad, y para facilitar su integración a sistemas existentes.

En esta sección se describe el modelo de contexto diseñado y se presentan las definiciones de los conceptos involucrados, constituyendo una de las contribuciones de esta tesis. La conceptualización completa de los elementos del modelo de contexto (conceptos, atributos, relaciones y axiomas estructurales) son presentadas en el Apéndice A. Estas definiciones son incorporadas luego a la *ontología de métricas e indicadores* [Martín & Olsina, 2003; Olsina & Martín, 2004] para satisfacer el requerimiento de una base ontológica para el marco C-INCAMI (dichas definiciones no constituyen una formalización ontológica estricta ya que no es el objetivo principal de esta tesis).

Cabe aclarar que el modelo presentado en la Figura 4.4 incluye los conceptos y relaciones básicos que describen cómo se representa la información de contexto, dejando los detalles que requieren la revisión completa de los conceptos de marco C-INCAMI, tratados en este capítulo, para el Capítulo 5. No obstante, todos los conceptos involucrados en la descripción de contextos y la contextualización de información de dominio están presentes en el modelo mencionado.

Como se describió en la Sección 3.3, el modelo de contexto se construye sobre los conceptos fundamentales de medición del marco INCAMI para reutilizar sus capacidades de cuantificación de atributos de entidades. Concretamente, el modelo de contexto extiende los conceptos *Entidad* y *Atributo* tal como se describirá a continuación.

El modelo de contexto propiamente dicho está compuesto de tres conceptos. El principal es Contexto (*Context*) que representa el estado relevante de la situación de una entidad.

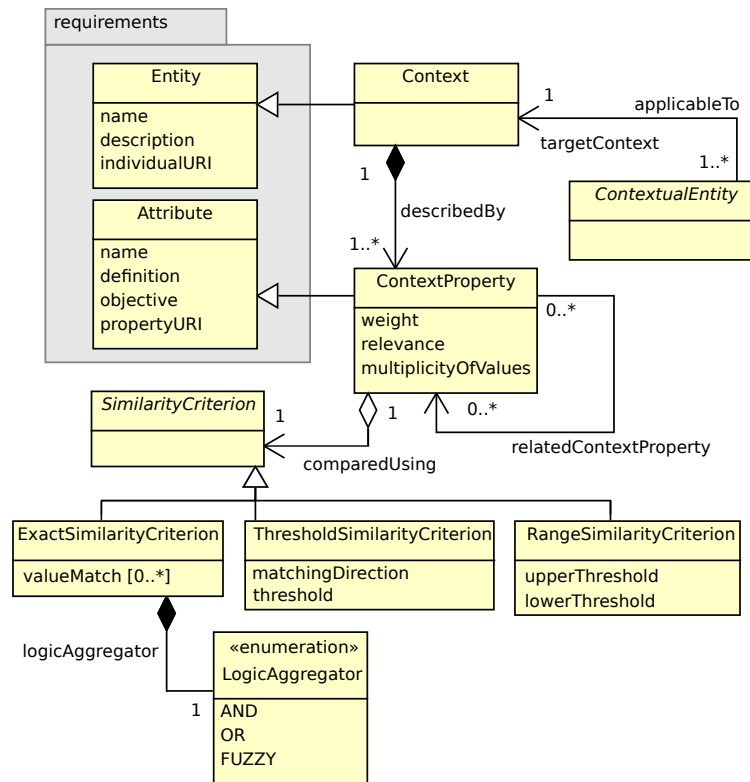


Figura 4.4: Diagrama de clases (UML) mostrando los conceptos y relaciones principales del módulo *context* del marco C-INCAMI.

DEFINICIÓN 7 (Context). Una entidad especial, descrita por un conjunto de propiedades de contexto que caracterizan el estado relevante de la situación de una entidad de interés dada. La situación de una entidad involucra la tarea y propósito para con esa entidad y sus relaciones con otras entidades respecto de dicha tarea y propósito. Un contexto puede ser caracterizado midiendo las propiedades (atributos) de las entidades involucradas en él.

Como puede observarse en la Figura 4.4, *Contexto* es considerado como un tipo especial de entidad en la cual se encuentran involucradas las entidades relevantes asociadas (establecido por la relación *relatedEntity*, mostrada en la Figura 4.3). Las entidades relevantes son aquellas cuyos atributos son relevantes en la interpretación de los datos relativos a la tarea y propósito para con la entidad de interés.

El contexto se describe mediante un conjunto de propiedades (*ContextProperty*), definidas como un tipo especial de *atributo*, que caracterizan a las entidades relevantes involucradas en el contexto de la entidad de interés. De esta forma, las *propiedades de contexto* pueden ser apropiadamente especificadas y cuantificadas de la misma forma que lo son los atributos especificados con propósitos de medición y evaluación, tal como se describe en la siguiente sección.

DEFINICIÓN 8 (ContextProperty) Un atributo que describe el *Contexto* de una entidad dada; este atributo pertenece a (la categoría de) alguna de las entidades que participan en el *Contexto* descripto.

Una propiedad de contexto está descrita por los metadatos heredados de la clase *Atributo* –*nombre*, *definición* y *objetivo*– y también por los metadatos propios:

- peso (*weight*): valor que refleja la importancia relativa de la propiedad dentro de la descripción del contexto correspondiente; los posibles valores se encuentran en un rango entre 0.0 y 1.0.
- relevancia (*relevance*): indica la pertinencia de usar la propiedad para describir el contexto de una entidad particular.

- multiplicidad de valores (*multiplicityOfValues*): indica la cantidad de valores (mediciones) que pueden ser asignados a la propiedad de contexto

Por su definición, estos tres atributos están pensados para asumir un valor cuando la propiedad de contexto es usada para describir una instancia particular de un contexto: el *peso* es relativo a todas las propiedades usadas para describir un contexto, la *relevancia* depende del caso y razón de porqué la propiedad es relevante para ese contexto, y la *multiplicidad de valores* depende tanto de la definición de la propiedad de contexto como del tipo de situación que se esté describiendo (la utilidad particular de cada uno de estos atributos, así como los aspectos relacionados a la cuantificación de las propiedades de contexto serán presentados, en el Capítulo 5, cuando se describan los detalles de la aplicación del enfoque sensible al marco de M&E C-INCAMI). Adicionalmente, el modelo permite registrar las propiedades de contexto relacionadas entre sí (relación *relatedContextProperty*). Dos o más propiedades de contexto pueden estar relacionadas entre sí cuando fueran necesarias para fundamentar o explicar apropiadamente un determinado aspecto de una situación. Por ejemplo, en una evaluación de *Calidad en Uso* de una cierta aplicación web, donde se ha llevado a cabo una prueba con usuarios reales en condiciones reales, las propiedades de contexto *Available bandwidth*, *Server load* y *Time frame* son necesarias en conjunto para explicar y/o caracterizar aspectos de rendimiento y/o efectividad; en una evaluación de calidad de contenidos, la propiedad *Supported natural language* está relacionada a las propiedades *Web site owner’s country* y *Target geographic region* ya que la primera depende de las últimas y, a la vez, pueden determinar los valores de la primera. Esta relación entre propiedades de contexto puede ser utilizada para descubrir propiedades relevantes al momento de describir contextos.

El modelo también permite especificar el contexto ideal en el cual una entidad del dominio de aplicación sensible al contexto puede ser aplicada o utilizada de forma coherente, basado en la naturaleza y definición de esa entidad. Estas entidades son llamadas Entidades Contextuales (*ContextualEntity*).

DEFINICIÓN 9 (ContextualEntity) Una entidad cuyo uso o interpretación correcta es sensible al contexto en el cual será aplicada y/o analizada.

Así, extendiendo –estableciendo relaciones de herencia con– el concepto *ContextualEntity* es posible contextualizar elementos de información del dominio de aplicación, asociando descripciones de contexto a los mismos mediante la relación *applicableTo* (como se observa en la Figura 4.4). Esto es posible debido al formalismo de modelado y representación elegido (presentado en el Capítulo 3, Sección 3). Volviendo a los conceptos del módulo *requirements* (descrito previamente y representado en la Figura 4.3) se identifican dos entidades contextuales a saber: *Modelos de Concepto* y *Atributos* (ver Figura 4.5).

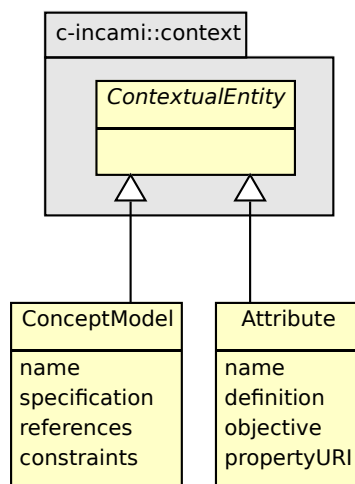


Figura 4.5: Diagrama de clases (UML) mostrando las entidades contextuales del módulo *requirements* del marco C-INCAMI.

Un *Modelo de Concepto* representa el concepto foco de la evaluación en términos de otros conceptos y, finalmente, atributos observables en la entidad correspondiente, respondiendo a la necesidad de información enunciada –que incluye la descripción del contexto relevante a la misma. Por lo tanto, el modelo de concepto debiera describir las propiedades deseables del contexto donde es aplicable. Por ejemplo, un modelo de concepto que representa *calidad* en función de características tales como *desempeño* y *confiabilidad* (entre otras) es aplicable para evaluar sistemas de tiempo real –como en el caso de sistemas de monitoreo de pacientes humanos o sistemas de navegación aeronáutica en los que una falla puede ocasionar pérdidas económicas o incluso humanas; un modelo de concepto de calidad que incluya el concepto de *accesibilidad* –respondiendo a las guías y técnicas desarrolladas por el *World Wide Web Consortium (W3C)* en el marco de la *Iniciativa de Accesibilidad Web*¹ (*WAI*)– es aplicable para la evaluación de aplicaciones web gubernamentales o que aspiran a satisfacer este tipo de estándares; un modelo de concepto que incluya el concepto de *mantenibilidad* es aplicable a una evaluación desde el punto de vista de un ingeniero de software, y no así para el punto de vista de un usuario final. De forma similar, un *Atributo* puede o no ser apropiado para evaluar una determinada entidad dependiendo del contexto relevante a la necesidad de información. Por ejemplo, en la evaluación de calidad de una aplicación web, la inclusión del atributo *Soporte de Lenguaje Extranjero* (o Internacionalización) [Calero *et al.*, 2004] en el modelo de concepto sólo tiene sentido cuando la aplicación está destinada a un audiencia global, tales como agencias de noticias internacionales o sitios de comercio electrónico con envíos internacionales, no así los sitios web que apuntan a comunidades enfocadas geográficamente (tal como el sitio web de los cines ODEON² que sólo se encuentran en Reino Unido).

Para llevar a cabo la comparación de descripciones de contexto, según lo propuesto en la Subsección 3.3.4, el módulo incluye un conjunto de conceptos que definen diferentes patrones de comparación o similitud entre propiedades de contexto. Concretamente, una propiedad de contexto tiene asociado (*comparedUsing*) un *Criterio de Similitud* (*SimilarityCriterion*) utilizado para determinar un valor de igualdad en la comparación de los valores asignados a dicha propiedad en un contexto dado con los valores de la misma propiedad especificada en otro contexto –a una misma propiedad especificada en diferentes contextos las llamaremos *propiedades de contexto equivalentes*. En este punto cabe aclarar que la comparación se lleva a cabo tomando una de las descripciones de contexto como referencia determinando la similitud de la segunda respecto de la primera. Los diferentes patrones utilizados ofrecen un enfoque más flexible a la simple comparación de valores uno a uno (tal como se propone en [Grossniklaus & Norrie, 2007]), considerando además que cada propiedad de contexto puede asumir más de un valor en la descripción de un contexto particular (como también se introdujo en la Subsección el 3.3.3). El modelo diseñado define tres tipos de criterios de similitud:

- *Criterio de Similitud de Umbral* (*ThresholdSimilarityCriterion*), establece un valor de umbral a partir del cual se determina un valor de similitud dependiendo de si el valor comparado de la propiedad equivalente se encuentra sobre o debajo de dicho umbral.
- *Criterio de Similitud de Rango* (*RangeSimilarityCriterion*), establece un rango (valor de umbral inferior y superior) a partir del cual se determina un valor de igualdad dependiendo de si el valor comparado de la propiedad equivalente se encuentra dentro del rango especificado.
- *Criterio de Similitud Exacto* (*ExactSimilarityCriterion*), establece un conjunto de valores (*value-Match*) y un operador de agregación lógico (*logicAggregator*) a partir de los cuales se determina un valor de similitud dependiendo de la agregación lógica de la igualdad uno a uno de los valores especificados en el criterio contra los valores comparados de la propiedad equivalente. El criterio de similitud exacto puede utilizar tres posibles operadores lógicos para realizar la agregación de

¹<http://www.w3.org/WAI/>

²<http://www.odeon.co.uk>

cada uno de los valores de similitud: AND, OR o FUZZY (el uso de cada uno de estos operadores será explicado en detalle en el Capítulo 5).

Los valores especificados por cada tipo de criterio de similitud (*threshold*, *upperThreshold*, *lowerThreshold* y *valueMatch*) corresponden a valores estimados o esperados obtenidos mediante el mecanismo de medición del marco INCAMI, cuyos conceptos se describen en la siguiente sección. Los aspectos relacionados a la definición de los criterios de similitud serán ampliados en el Capítulo 5.

DEFINICIÓN 10 (SimilarityCriterion) Patrón de valoración que permite determinar la similitud semántica entre los valores especificados para una misma propiedad de contexto en diferentes descripciones de contexto.

DEFINICIÓN 11 (ThresholdSimilarityCriterion) Criterio de similitud basado en un patrón de umbral para determinar la similitud semántica entre los valores de una propiedad de contexto en diferentes descripciones de contexto.

DEFINICIÓN 12 (RangeSimilarityCriterion) Criterio de similitud basado en un patrón de rango para determinar la similitud semántica entre los valores de una propiedad de contexto en diferentes descripciones de contexto.

DEFINICIÓN 13 (ExactSimilarityCriterion) Criterio de similitud basado en un patrón de equivalencias uno a uno agregadas de forma lógica (mediante un agregador lógico) para determinar la similitud semántica entre los valores de una propiedad de contexto en diferentes descripciones de contexto.

DEFINICIÓN 14 (LogicAggregator) Operador lógico utilizado por un criterio de similitud exacto para agregar en un único valor los resultados parciales de cada equivalencia uno a uno.

La decisión de qué criterio de similitud se utiliza para una propiedad de contexto depende del caso para el cual se describa el contexto –para describir el contexto de una situación real o de una entidad contextual. Este tema será tratado con detalle en el Capítulo 5.

4.4. Definición e Implementación de la Medición

El módulo *measurement* incluye los metadatos de los conceptos y relaciones destinados a especificar el diseño e implementación de la medición (ver Figura 4.6). Estas actividades son gestionadas por un *Proyecto de Medición (MeasurementProject)*, como un subproyecto del *Proyecto de M&E* principal. En el Proyecto de Medición se especifican las entidades concretas (*Entity*) que serán medidas así como las métricas seleccionadas para cuantificar sus atributos.

DEFINICIÓN 15 (MeasurementProject) *Un proyecto que permite, partiendo desde una especificación de requerimientos no funcionales, seleccionar las métricas apropiadas y registrar los valores en un proceso de medición.*

El diseño de la medición consiste concretamente en un conjunto de *Métricas (Metric)* especificadas y seleccionadas apropiadamente para cuantificar cada uno de los atributos involucrados en la especificación de requerimientos no funcionales. En general un atributo puede ser cuantificado por más de una métrica, pero en un proyecto de medición concreto sólo una es asignada para cada atributo. Una métrica provee una especificación de cómo cuantificar un atributo particular de una entidad siguiendo la especificación de un *Método (Method)*, y de cómo representar sus valores usando una *Escala (Scale)* particular. Una escala se caracteriza por su tipo (*ScaleType*), que describe la naturaleza de la relación entre los valores de la misma –pudiendo ser *nominal*, *ordinal*, *intervalo*, *proporción* o *absoluta*–, y por

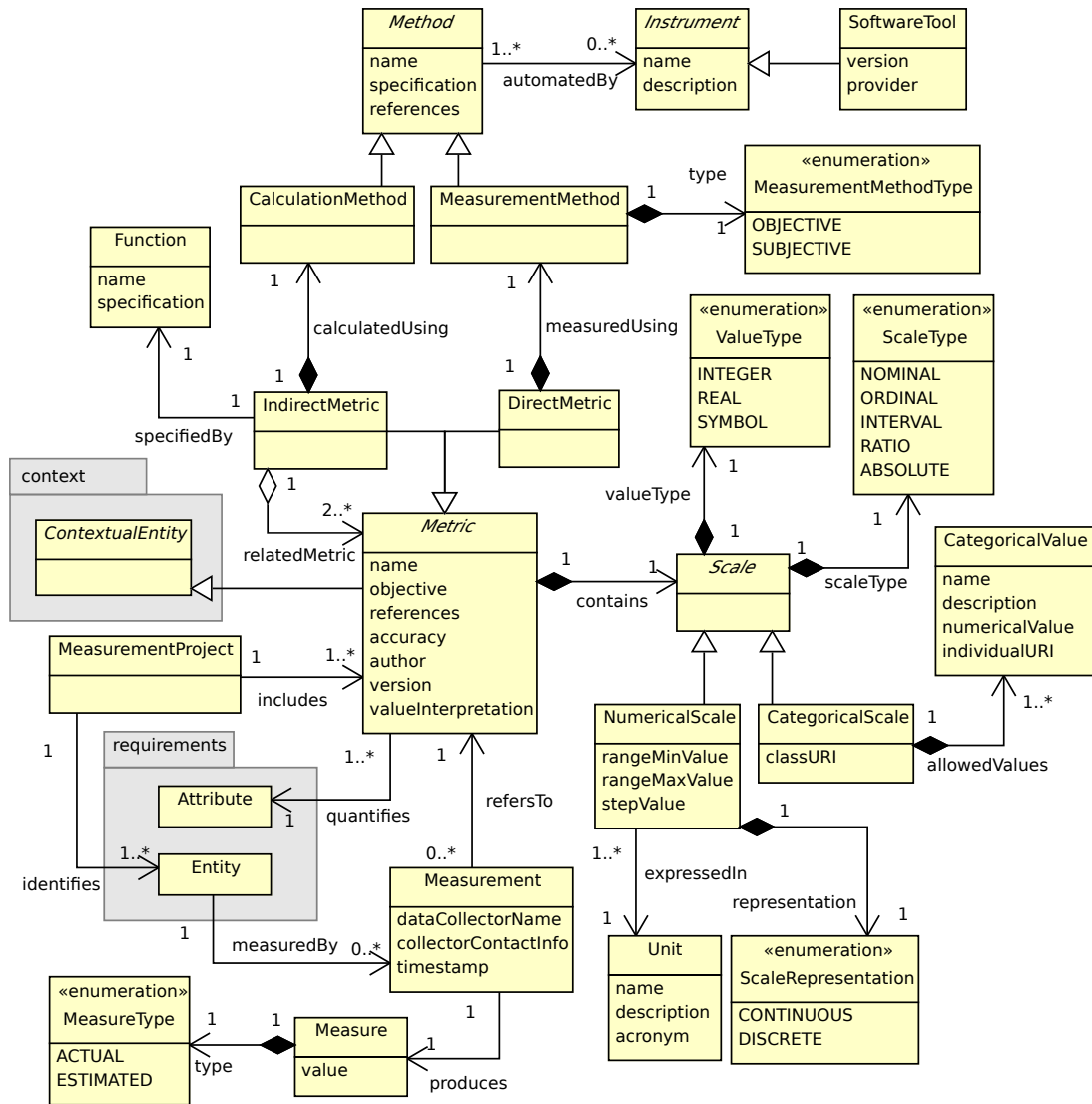


Figura 4.6: Diagrama de clases (UML) mostrando los conceptos y relaciones del módulo *measurement* del marco C-INCAMI.

el tipo de los valores de la misma (*ValueType*) –tales como *entero*, *real* o *símbolo* [Fenton & Pfleeger, 1997; Zuse, 1998]. Adicionalmente, el marco considera dos clases de escalas más específicas. Una *Escala Categórica* (*CategoricalScale*) describe un conjunto de categorías (valores de tipo *SYMBOL*) que pertenecen a la escala (*CategoricalValue*), cada una de las cuales posee, además de su *nombre*, una *descripción* y un *valor numérico* utilizado para establecer el orden de la categoría (cuando el tipo de escala fuera *ORDINAL*). Por otro lado, una *Escala Numérica* (*NumericalScale*) representa un conjunto de valores numéricos siguiendo una representación (*ScaleRepresentation*) *discreta* o *continua* (de tipo *INTEGER* o *REAL* respectivamente), en un rango (*rangeMinValue*, *rangeMaxValue*) e intervalo indicado (*stepValue*), si correspondiera, expresados en alguna *Unidad* (*Unit*) definida. Para cada métrica también se registra de forma textual una descripción de cómo interpretar los valores generados por la métrica (*valueInterpretation*), un valor que describa de forma aproximada la precisión con que la métrica puede generar el valor real del atributo (*accuracy*), una descripción textual del objetivo por el cual se aplica la métrica (*objective*), el nombre del autor de la métrica (*author*) y la/s referencia/s (*references*) –si las hubiera- de dónde se obtuvo la información para especificar la métrica. Finalmente, y considerando que un atributo puede ser cuantificado por más de una métrica, es posible indicar un valor de *versión* para la misma considerando que la experiencia puede llevar a modificar las métricas ya utilizadas. De esta forma, las métricas aplicadas originalmente permanecen sin cambios para evitar

problemas de consistencia entre mediciones tomadas a lo largo del tiempo. Todos estos atributos pretenden describir la métrica de forma que su definición sea lo más clara y objetiva posible, facilitando la repetibilidad, trazabilidad y reuso de los resultados.

DEFINICIÓN 16 (Metric) *El método de medición o cálculo y la escala de medición.*

DEFINICIÓN 17 (Method) *(sinon. procedimiento) Secuencia lógica de operaciones y posibles heurísticas, especificadas de forma genérica, para permitir la realización de la descripción de una actividad.*

DEFINICIÓN 18 (Scale) *Un conjunto de valores con propiedades definidas.*

DEFINICIÓN 19 (CategoricalScale) *Una escala donde los valores medidos o calculados son categorías y no pueden ser expresadas en unidades, en un sentido estricto.*

DEFINICIÓN 20 (NumericalScale) *Una escala donde los valores medidos o calculados son números que pueden ser expresados en unidades, en un sentido estricto.*

DEFINICIÓN 21 (Unit) *Una cantidad particular definida y adoptada por convención, con la cual otras cantidades del mismo tipo son comparadas para expresar su magnitud relativa a esa cantidad.*

El marco distingue dos tipos de métricas. Las *Métricas Directas (DirectMetric)* son aquellas cuyos valores son obtenidos directamente midiendo el atributo correspondiente a la entidad asociada, usando, en este caso, un *Método de Medición (MeasurementMethod)* el cual puede ser *objetivo* o *subjetivo*. Por otro lado, las *Métricas Indirectas (IndirectMetric)* permiten calcular el valor de un atributo a partir de los resultados de otras métricas (directas o indirectas) siguiendo la especificación de una *Función (Function)* y un *Método de Cálculo (CalculationMethod)* particular. Adicionalmente el marco permite registrar los metadatos de los *Instrumentos (Instrument)* que pudieran aplicarse para implementar los métodos utilizados por las métricas. En particular se contempla la especificación de *Herramientas de Software (SoftwareTool)* como instrumento para automatizar dichos métodos.

DEFINICIÓN 22 (DirectMetric) *(sinon. métrica base o simple) Una métrica de un atributo que no depende de la métrica de algún otro atributo.*

DEFINICIÓN 23 (MeasurementMethod) *(sinon. regla de conteo, protocolo) La secuencia lógica de operaciones y posibles heurísticas especificadas para permitir la realización de la descripción de una métrica directa mediante una medición.*

DEFINICIÓN 24 (IndirectMetric) *(sinon. métrica derivada o híbrida) Una métrica de un atributo que se deriva a partir de métricas de uno o más atributos.*

DEFINICIÓN 25 (Function) *Algoritmo o fórmula que combina dos o más métricas.*

DEFINICIÓN 26 (CalculationMethod) *La secuencia lógica de operaciones especificadas para permitir la realización de la descripción de una fórmula o indicador mediante un cálculo.*

DEFINICIÓN 27 Instrument *Dispositivo físico o virtual que permite asistir en la obtención del valor formal que describe una propiedad empírica de una entidad en el mundo real.*

DEFINICIÓN 28 SoftwareTool *(sinon. instrumento de software) Es una aplicación de software que automatiza parcialmente o totalmente un método de medición o cálculo.*

Para representar los resultados de la implementación de la medición se provee el concepto de *Medición* (*Measurement*) que representa la actividad misma a partir de la cual se genera el valor de la *Medida* (*Measure*), resultante de la aplicación de la métrica correspondiente. También se especifica para una *Medición* los metadatos asociados tales como el nombre del responsable de llevar a cabo la medición e información de contacto (*dataCollectorName* y *collectorContactInfo*) y el instante de tiempo (*timestamp*) en que fue tomada. Adicionalmente es posible indicar si el valor de una medida corresponde a un valor *real* o *estimado* (*MeasureType*) permitiendo al modelo no solo registrar valores de mediciones reales sino también valores estimados para luego realizar comparaciones que permitan determinar desvíos y efectuar correcciones para estimaciones futuras. Finalmente, todas las mediciones son asociadas a la entidad sobre la cual fueron llevadas a cabo (mediante la relación *measuredBy*). Es posible registrar más de una *medición* para cada *métrica* que cuantifica un *atributo*, incluso en un mismo proyecto de M&E. Esto permite cuantificar métricas indirectas basadas en la agrupación de valores provenientes de otras métricas, tal como en el cálculo de sumatorias.

DEFINICIÓN 29 (Measurement) *Una actividad que usa la definición de una métrica para producir el valor de una medida.*

DEFINICIÓN 30 (Measure) *El número o categoría asignada a un atributo de una entidad luego de llevar a cabo una medición.*

Respecto del enfoque sensible al contexto utilizado en el marco C-INCAMI deben considerarse los conceptos del módulo *measurement* que sean sensibles al contexto. Como se indicó previamente, el diseño de la medición se lleva a cabo mediante el diseño/selección de métricas para cuantificar los atributos que componen los requerimientos no funcionales a medir y evaluar. También se recuerda que la definición de una métrica incluye un método específico para llevar a cabo la medición correspondiente. La aplicación de dicho método depende de la disponibilidad de un conjunto de recursos, tanto tecnológicos, de conocimiento, entre otros. Estos aspectos pueden ser observados en el contexto real de la entidad bajo análisis. Adicionalmente, la naturaleza de la métrica (evidente en su *objetivo*, *interpretación del valor*, o en la especificación de su *método*) puede estar asociada a determinados aspectos del atributo a medir, particulares del contexto de aplicación. Por ejemplo, la métrica *Número de Clases* puede ser utilizada para cuantificar el atributo *Tamaño de Código Fuente* pero sólo cuando se aplique a una pieza de software desarrollada con tecnología orientada a objetos. Por todo esto, se considera que las métricas son sensibles al contexto por lo que el concepto *Métrica* del módulo *measurement* es definido como una *Entidad Contextual* (del módulo *context* presentado en la sección 4.3.1) tal como se observa en la Figura 4.6. De esta forma es posible asignar a una métrica la descripción del contexto en el cual es aplicable.

Repasando los conceptos (y definiciones correspondientes) vistos en el módulo *measurement*, cabe destacar que la disponibilidad de contar con los metadatos que describen cada uno de los conceptos mencionados permite especificar de forma clara y explícita las métricas que definen las tareas a realizar para llevar a cabo la medición de las entidades correspondientes a partir de los requerimientos no funcionales especificados mediante el módulo *requirements*. Más aún, la especificación de las métricas (y sus elementos) pueden ser almacenadas en repositorios comunes a todos los proyectos de la organización, promoviendo el reuso de las mismas y la consistencia de las mediciones realizadas que permiten llevar a cabo comparaciones entre diferentes proyectos.

4.5. Definición e Implementación de la Evaluación

El módulo *evaluation* incluye los metadatos de los conceptos y relaciones necesarios para especificar el diseño y la implementación de la evaluación (ver Figura 4.7). Estas actividades son gestionadas por un *Proyecto de Evaluación* (*EvaluationProject*), que conforma parte del *Proyecto de M&E* principal,

que describen el nivel de satisfacción correspondiente, un *modelo* que especifica la fórmula o función utilizada para obtener dicho valor y un *método de cálculo* (el mismo definido en la sección anterior) que describe cómo se implementa el modelo mencionado. De forma análoga a las métricas, un indicador especifica la precisión con la que estima una preferencia o requerimiento, referencias a la fuente de la cual pudo haber sido obtenida su especificación, el autor y la versión del indicador (ya que el mismo indicador podría ser diseñado de forma diferente por otro autor).

Se distinguen dos tipos de indicadores. Por un lado, un *Indicador Elemental* (*ElementaryIndicator*) evalúa requerimientos de bajo nivel, es decir, *Atributos* combinados en un *Modelo de Concepto*. Cada *indicador elemental* tiene asociado un *Modelo Elemental* (*ElementaryModel*) que define una función que interpreta (convierte) las medidas obtenidas a partir de la *Métrica* que cuantifica el *Atributo* interpretado (el dominio) a valores expresados en la *Escala* del indicador (el rango). Esta función puede definir una conversión de valores de la métrica –en cuanto a su cardinalidad– de *muchos-a-uno* (el modelo contempla la existencia de más de un valor de medición como entrada, por ejemplo $x \supset \{“inglés”, “español”\}$) o *uno-a-uno* (el modelo utiliza un único valor de medición como entrada, por ejemplo $x = 4$); en este último caso, ya que una métrica puede registrar un número arbitrario de medidas para el atributo correspondiente, se entiende que el evaluador deberá elegir –mediante el mecanismo adecuado, que pudiera ser provisto por la herramienta que de soporte al marco– cuál de las mediciones será utilizada como entrada para el cálculo del indicador elemental correspondiente. Los *modelos elementales* que pueden asignarse a un *indicador elemental* para interpretar un *atributo* determinado son aquellos que interpretan (relación *interprets*) la *métrica* que ha sido asignada para cuantificar el atributo en cuestión, ya que la especificación del modelo elemental se realiza en función de un conjunto de valores representados en una escala particular, que debe ser la misma que utiliza la métrica que cuantifica el atributo evaluado por el indicador elemental.

Por otro lado, un *Indicador Global* (*GlobalIndicator*) evalúa e interpreta requerimientos de nivel medio y alto, es decir, los *Conceptos Calculables* incluidos en un *modelo de concepto*. Un indicador global combina (relación *relatedIndicators*) indicadores globales y elementales en una estructura jerárquica que iguala al modelo de concepto interpretado. A los indicadores globales incluidos en un nivel intermedio de dicha jerarquía se los denomina también *indicadores parciales*. Un *indicador global* tiene asociado un *Modelo Global* (*GlobalModel*) o de agregación que especifica cómo combinar los valores de los indicadores elementales y globales asociados –aquellos que interpretan los conceptos y atributos que combina el concepto calculable interpretado por el indicador global– para obtener un valor que represente el nivel de satisfacción del concepto calculable interpretado. Las conversiones de un modelo global son siempre uno-a-uno, ya que toman como entrada valores producidos por modelos elementales (o globales) que producen un único resultado como salida. Diferentes modelos de agregación pueden ser usados para llevar a cabo las evaluaciones, dependiendo del comportamiento deseado para la misma. Por ejemplo, el *Modelo LSP* (*Logic Scoring of Preference*) [Dujmovic, 1996], que permite modelar relaciones entre requerimientos de forma flexible aplicando diferentes operadores lógicos que van desde la disyunción pura hasta la conjunción pura –descrito en la Sección 3.2.3.1. También otros modelos pueden ser usados tales como redes neuronales o lógica difusa, entre otros. Considerando que pueden utilizarse diferentes *modelos globales*, es posible especificar para cada uno de ellos, el conjunto de operadores (atributo *operators*) disponibles para ser utilizados (por ejemplo, el *Modelo Aditivo* posee un único operador –suma– mientras el *Modelo LSP* provee los operadores presentados en la Figura 3.4). Consecuentemente, cada indicador global debe especificar el operador (atributo *operator*) seleccionado para agregar los valores provenientes de los indicadores relacionados. Así también, todo indicador debe especificar su peso relativo (atributo *weight*) al resto de los indicadores que conforman un mismo nivel de agregación. Este peso, al igual que el operador, será utilizado por los modelos globales que requieran este parámetro durante el cálculo de valores de indicadores globales/parciales. Tanto los modelos globales como elementales tienen asociado un conjunto de *Criterios de Decisión* (*DecisionCriteria*) que asignan una interpretación a los valores de la escala de los indicadores en base a un

rango especificado (umbrales inferior y superior) permitiendo analizar el nivel de satisfacción alcanzado por cada requerimiento no funcional, tanto global como elemental –conceptos calculables y atributos respectivamente.

DEFINICIÓN 32 (Indicator) (*sinon. criterio, preferencia*) *El método de cálculo, la escala, el modelo y criterios de decisión para proveer una estimación o valuación de un requerimiento no funcional respecto de necesidades de información definidas.*

DEFINICIÓN 33 (ElementaryIndicator) (*sinon. preferencia elemental, criterio elemental*) *Un indicador que evalúa un atributo sin depender de otros indicadores.*

DEFINICIÓN 34 (ElementaryModel) (*sinon. función de criterio elemental*) *Algoritmo o función, con criterios de decisión asociados, que modela a un indicador elemental.*

DEFINICIÓN 35 (DecisionCriteria) (*sinon. niveles de aceptabilidad*) *Umbrales o patrones que describen el nivel de aceptabilidad en un resultado dado que pueden utilizarse para determinar una necesidad de acción o investigación posterior.*

DEFINICIÓN 36 (DecisionCriterion) (*sinon. nivel de aceptabilidad*) *Rango que define un umbral o patrón que compone un conjunto de criterios de decisión.*

DEFINICIÓN 37 (GlobalIndicator) (*sinon. preferencia global, criterio global, indicador parcial*) *Un indicador que es derivado de otros indicadores para evaluar o estimar un concepto calculable.*

DEFINICIÓN 38 (GlobalModel) (*sinon. modelo de agregación, modelo o función de valuación*) *Algoritmo o función, con criterios de decisión asociados, que modela un indicador global.*

Respecto de la implementación de la evaluación, el módulo incluye el concepto de *Evaluación (Evaluation)* que representa la actividad propiamente dicha, que utiliza la especificación de un indicador, ya sea elemental o global, para efectuar el cálculo correspondiente. Dicha actividad produce un *Valor de Indicador (IndicatorValue)* que representa el valor de preferencia correspondiente al componente de los requerimientos no funcionales interpretado, ya sea un atributo o un concepto calculable. El *valor* producido para dicho *indicador* es interpretado utilizando los *criterios de decisión* asociados al *modelo* –elemental o global- del mismo. La entrada a la evaluación de un *indicador elemental* es el valor medido del *atributo* interpretado. Como se mencionó en la sección anterior, múltiples mediciones pueden ser registradas para cuantificar un atributo. No obstante, sólo un resultado de medición del atributo es utilizado como entrada para la interpretación de atributos mediante indicadores elementales. Por otro lado, las entradas a la evaluación de un *indicador global* son los valores de indicador obtenidos de las evaluaciones elementales y/o parciales relacionadas. Finalmente, cada evaluación se asocia a la entidad partir de la cual se obtuvieron las medidas para realizar la evaluación.

DEFINICIÓN 39 (Evaluation) (*sinon. cómputo*) *Actividad que usa la definición de un indicador para producir un valor de indicador.*

DEFINICIÓN 40 (Indicator Value) (*sinon. valor de preferencia*) *El número o categoría asignada a un concepto calculable mediante la ejecución de un cálculo.*

De esta forma, el uso de indicadores, tal como se define en el marco C-INCAMI, permite implementar el mecanismo de interpretación de las necesidades de información planteado por el enfoque GQM [Basili *et al.*, 1994], que va desde el nivel inferior o cuantitativo –las métricas- hasta el nivel superior o conceptual –el concepto calculable foco de la evaluación-, pasando por los niveles intermedios u operativo –la jerarquía de subconceptos del modelo de concepto. Gracias al modelo global o de agregación definido para los indicadores globales es posible llevar a cabo un proceso de agregación cuantitativo

paso a paso que sigue la estructura jerárquica definida por el modelo de concepto desde abajo hacia arriba calculando un valor de preferencia para todos los indicadores parciales (globales) para obtener finalmente el valor global que representa el nivel de satisfacción de los requerimientos no funcionales enunciados por la necesidad de información.

Al igual que en la medición, la evaluación de requerimientos no funcionales puede verse afectada por el contexto relevante a la necesidad de información. Es así que, a continuación, se identifican los conceptos del módulo *evaluation* que son sensibles al contexto y, por lo tanto, especializaciones del concepto *ContextualEntity* del módulo *context*. Como se mencionó al principio de la sección, la evaluación se especifica mediante el diseño de *indicadores* que incluyen modelos utilizados para calcular los valores de preferencia de los elementos que componen un *modelo de concepto* (*conceptos calculables* y *atributos*). Estos modelos incluyen además los *criterios de decisión* que permiten llegar a una interpretación de dichos valores de preferencia en función del nivel de satisfacción de los requerimientos no funcionales. Intuitivamente, y de forma análoga a lo visto en el módulo de medición, los *indicadores* serían sensibles al contexto. No obstante, el comportamiento esencial de los indicadores es determinado por los *modelos –elementales y globales–* los cuales determinan el *método de cálculo* y la *escala* correspondiente para el *indicador*. Por lo tanto, son los *modelos elementales y globales* los conceptos del módulo *evaluation* definidos como *entidades contextuales* (ver Figura 4.7). En primer lugar, un *modelo elemental*, que define la conversión de los valores de las métricas a una escala normalizada, puede depender de ciertas propiedades del contexto en el que se aplican e interpretan los resultados de utilizar el indicador correspondiente. Por ejemplo, considere el atributo *Soporte de Lenguaje Extranjero*, perteneciente a la categoría de entidad *Sitio Web*, cuantificado por la métrica *Lenguajes Naturales Soportados* –que asigna valores de una escala categórica que enumera diferentes lenguajes naturales. La elección de un modelo elemental para el indicador que interprete este atributo debe realizarse en función de las propiedades de contexto *Región Geográfica Objetivo* y *País del Propietario del Sitio Web*. Un modelo elemental que realiza la evaluación considerando los lenguajes “español” y “portugués” (ver Ecuación 4.1) es aplicable cuando la *Región Geográfica Objetivo* del sitio web es “América del Sur” y el *País del Propietario del Sitio Web* sea, por ejemplo, “Argentina”. Cabe notar que cuando sólo el lenguaje correspondiente al *Propietario del Sitio Web* se encuentra disponible como opción en el sitio el modelo elemental asigna un valor de preferencia 0.

$$EM_{LNS1} = \begin{cases} 0 & \text{si "español"} \in M_{LS} \\ 100 & \text{si "español"} \in M_{LS} \wedge \text{"portugués"} \in M_{LS} \end{cases} \quad (4.1)$$

Otro modelo elemental que realiza la evaluación considerando los lenguajes “español”, “portugués” e “inglés” (ver Ecuación 4.2) es aplicable cuando la *Región Geográfica Objetivo* es “América del Sur” y “América del Norte” y el mismo *País del Propietario del Sitio Web* que en el caso anterior.

$$EM_{LNS2} = \begin{cases} 0 & \text{si "español"} \in M_{LS} \\ 70 & \text{si "español"} \in M_{LS} \wedge \text{"portugués"} \in M_{LS} \\ 100 & \text{si "español"} \in M_{LS} \wedge \text{"portugués"} \in M_{LS} \wedge \text{"inglés"} \in M_{LS} \end{cases} \quad (4.2)$$

Alternativamente otro modelo elemental podría especificar a “Europa” como *Región Geográfica Objetivo* cuando incluya en su definición lenguajes como “italiano” y “alemán” –entre otros.

En segundo lugar, un modelo global, que define el mecanismo de agregación de los valores de preferencia elemental (proveniente de indicadores elementales) y parcial (proveniente de indicadores globales de nivel intermedio), depende principalmente de la disponibilidad de los recursos de cómputo necesarios para llevar a cabo el método de cálculo asociado, aspecto observable en el contexto real del proyecto de M&E. Finalmente, los criterios de decisión asociados tanto a los modelos globales como elementales pueden haber sido diseñados para satisfacer un determinado estándar o norma de

evaluación o certificación, tanto interno como externo (por ejemplo CMMI [CMMI Product Team, 2006]). Estos aspectos del contexto también pueden ser relevantes para la elección de los modelos globales o elementales.

Tal como se han descrito, los metadatos provistos por el módulo *evaluation* permiten especificar de forma explícita y cuantitativa la información que permite llegar a una clara y consistente interpretación de los resultados de la evaluación.

4.6. Integración del Marco C-INCAMI a la Organización

Como parte de las contribuciones de esta tesis, se ha incluido un mecanismo para conectar las especificaciones de algunos de los conceptos del marco C-INCAMI utilizados en el diseño e implementación de la medición y la evaluación, a los datos del dominio de aplicación que estuvieran almacenados en repositorios o memorias de proyectos de la organización. Este mecanismo facilitaría la incorporación e integración del marco C-INCAMI a los sistemas de la organización, mejorando a la vez la consistencia de las mediciones con el dominio de aplicación de la organización.

El mecanismo diseñado se basa en el uso de tecnologías de la Web Semántica³, tales como *Identificadores Uniformes de Recursos* o *URIs* (*Uniform Resource Identifier*) [Berners-Lee *et al.*, 2005], utilizados por los estándares XML [W3C-XML, 2012], RDF [W3C-RDF, 2004] y OWL [W3C-OWL, 2004] (entre otros), para identificar recursos en la web así como también otros recursos físicos o abstractos. En este sentido, y como restricción del mecanismo diseñado, se asume que la organización mantiene repositorios accesibles mediante URIs, tales como los repositorios semánticos *Sesame*⁴ y *OWLIM*⁵, en los que se almacenan tanto Esquemas RDF (*RDFS* o *RDF Schemas*) [W3C-RDFS, 2004], que definen el vocabulario del dominio de la organización, como datos RDF sobre la operatoria de negocios de la organización que conforman con dicho vocabulario (estas tecnologías se describen en el Cuadro 4.1). De esta forma, utilizando un atributo de tipo *URI* en los metadatos del concepto apropiado del marco C-INCAMI (recordar la Figura 4.3), es posible referenciar la definición del concepto correspondiente almacenado en los repositorios de la organización. Estos atributos de tipo *URI* mantienen referencias a diferentes elementos del vocabulario de *RDF* y *RDF Schemas*, según corresponda. Más aún, las definiciones de los conceptos del marco C-INCAMI podrían ser automáticamente recuperados y especificados a partir de su fuente

Cuadro 4.1. Tecnologías de la Web Semántica.

- ▷ Un **URI** (Uniform Resource Identifier) es una secuencia de caracteres que permite identificar de forma inequívoca y uniforme diferentes tipos de recursos (físicos o abstractos), pudiendo ser utilizados en diferentes contextos. La sintaxis URI permite definir diferentes esquemas de identificación, cada uno de ellos con reglas sintácticas, semánticas y mecanismos de acceso específicos. Un identificador URI se especifica en la forma “<nombre de esquema> : <parte específica del esquema>”, donde ésta última contiene información jerárquica que identifica al recurso (incluyendo un identificador de autoridad y un camino) y opcionalmente información adicional que incluye una consulta (como una secuencia de pares <clave>=<valor>) y un fragmento que identifica un recurso secundario como parte del principal.
- ▷ **RDF** (Resource Description Framework) es un modelo de datos abstracto que permite enunciar sentencias acerca de recursos –especialmente web–, compuestas de *sujeto*, *predicado* y *objeto*, llamadas *triplas RDF*. El *sujeto* indica a un recurso, el *predicado* indica alguna relación o propiedad del sujeto, y el *objeto* indica el recurso relacionado o el valor asignado al atributo para dicho sujeto. Los tres componentes de una sentencia RDF pueden ser recursos explícitamente declarados en RDF identificados mediante un URI. Una colección de sentencias RDF conforman un grafo múltiple dirigido y etiquetado, donde el sujeto y el objeto son los nodos y la relación los bordes (enlaces entre nodos). El modelo RDF puede ser serializado y codificado en distintos formatos. RDF constituye el componente principal de la web semántica para el almacenamiento, intercambio y uso de información legible por las computadoras distribuida en la web.
- ▷ **RDF Schema (RDFS)** es un lenguaje de descripción de vocabularios RDF, definido como una extensión semántica de RDF. Provee mecanismos para describir tipos de recursos (*rdfs:Class*) y las relaciones entre ellos (*rdfs:Property*). Las relaciones se describen en términos de las clases de recursos a las cuales se aplican (*rdfs:domain*) y el tipo de recurso al que se refieren (*rdfs:range*).

³<http://www.w3.org/2001/sw/>

⁴<http://www.openrdf.org/>

⁵<http://www.ontotext.com/owlim>

en los repositorios de la organización. Cabe aclarar que no es necesario ni obligatorio enlazar las especificaciones realizadas con el marco C-INCAMI a las definiciones del dominio de la organización. No obstante, esta capacidad puede resultar en una ventaja al momento de incorporar el soporte tecnológico del marco a los sistemas de la organización, o –en caso de no disponer de los repositorios con las características mencionadas– para proveer un punto de inicio para incorporar tecnología de la web semántica, con todos los beneficios que esta provee. Una de las principales ventajas de contar con estos repositorios semánticos (por ejemplo, repositorios RDF) consiste en la posibilidad de efectuar sobre los mismos consultas sobre los esquemas así como sobre las instancias de datos, permitiendo hacer inferencias acerca de información que no se encuentra explícitamente declarada en los mismos, tales como relaciones entre clases/instancias, propiedades, dominios, rangos, etc. En RDF esto es posible gracias al lenguaje de consulta RQL (*RDF Query Language*) [RQL-ICS-FORTH, 2008] –y derivados– que permite, al igual que una consulta SQL (*Structured Query Language*) en bases de datos relacionales, especificar consultas sobre el modelo de datos de RDF (triplas RDF –conformadas por *sujeto*, *predicado* y *objeto*) instanciando variables definidas sobre dicho modelo, correspondientes tanto a recursos como a propiedades RDF.

Los conceptos del marco C-INCAMI que pueden ser conectados a definiciones de la organización son –como se observa en la Figura 4.8–, del módulo *requirements*, las categorías de entidad, las entidades y sus atributos, y del módulo *measurement*, la escala categórica, sus valores categóricos y las unidades.

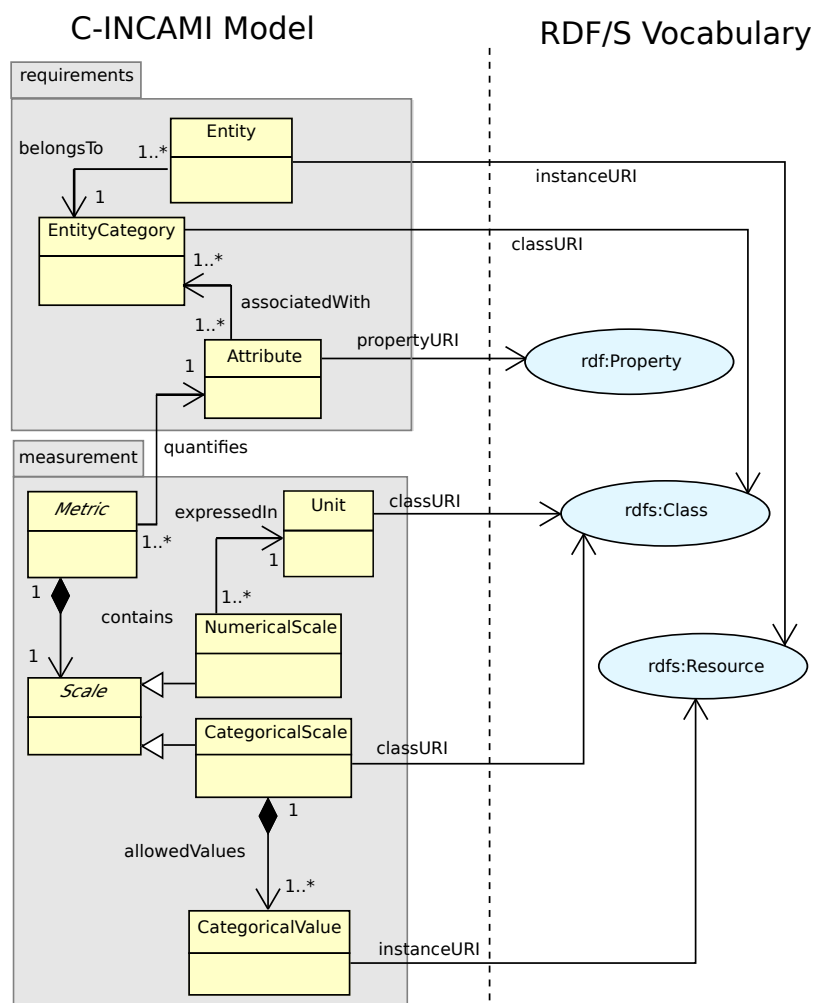


Figura 4.8: Conceptos del Marco C-INCAMI que se conectan al dominio de aplicación de la organización en repositorios RDF mediante direcciones URI.

En la Figura 4.9 se presenta un ejemplo donde un conjunto de instancias del marco C-INCAMI (representadas en un diagrama de objetos UML) referencian definiciones RDFS y RDF almacenadas en un repositorio de la organización (representadas en un grafo RDF). Las definiciones del repositorio de la organización del ejemplo son identificadas usando URIs en la forma de *nombres calificados* –compuestos por un prefijo que representa el URI de un espacio de nombres (el vocabulario del dominio) y un nombre local. De esta forma, el URI “*http://myorg.com/business-domain/SoftwareModule*” es abreviado como “*org:SoftwareModule*”, donde un prefijo de espacio de nombres (“*org:*”) reemplaza a la dirección URL “*http://myorg.com/business-domain/*”.

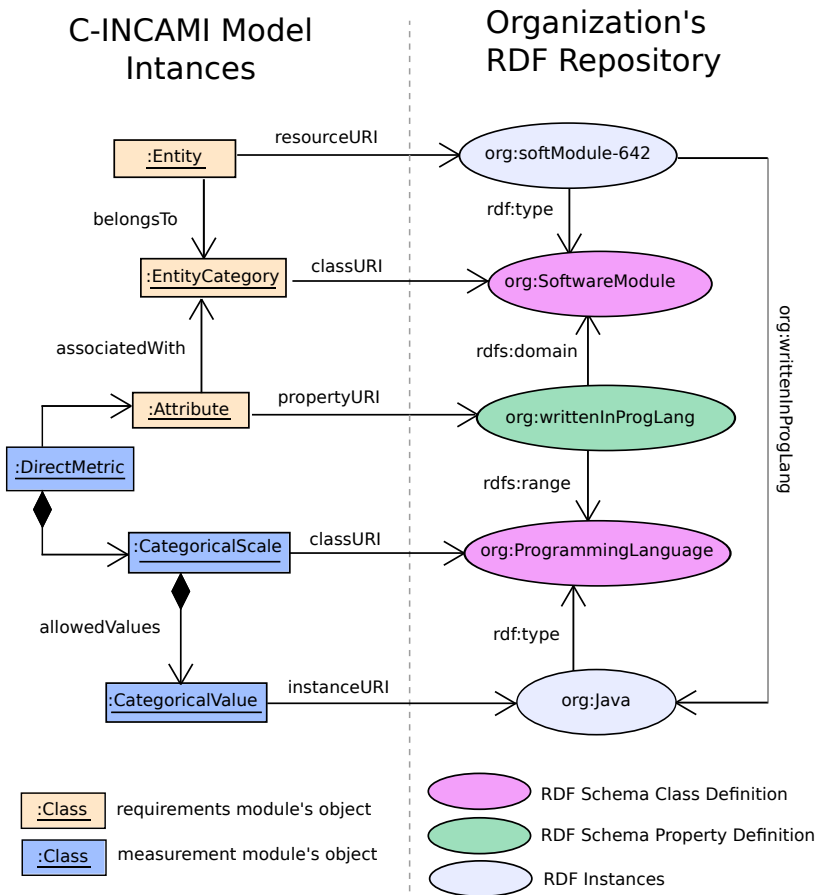


Figura 4.9: Ejemplo de especificaciones C-INCAMI (diagrama de objetos UML) que referencian definiciones en un repositorio RDF (grafo RDF).

Comenzando por el módulo *requirements*, una *Categoría de Entidad* (*EntityCategory*) mantiene una referencia (*classURI*) a una *Clase* del vocabulario de RDF Schemas (*rdfs:Class*) que contiene la definición de un concepto del vocabulario RDF del dominio de la organización, que corresponde a dicha categoría. En el ejemplo de la Figura 4.9, una *categoría de entidad* especificada en el marco C-INCAMI referencia la definición de la *clase* RDF “*org:SoftwareModule*” que define un *Módulo de Software*. Una *Entidad* (*Entity*) es enlazada (*instanceURI*) a un *Recurso RDF* (*rdfs:Resource*), definido en los datos RDF de la organización, que identifica una *instancia* RDF de alguna *clase* RDF referenciada por la *categoría de entidad* a la que pertenece dicha *entidad*. En el ejemplo mencionado, una *entidad* especificada en el marco C-INCAMI referencia la definición de la *instancia* RDF “*org:softModule-642*” cuyo tipo (*rdfs:type*) es la *clase* RDF “*org:SoftwareModule*” referenciada por la *categoría de entidad* a la que pertenece dicha *entidad*. Un *Atributo* (*Attribute*) es enlazado (*propertyURI*) a una *Propiedad* RDF (*rdf:Property*), definida en el vocabulario RDF del dominio de la organización, aplicable a (*rdfs:domain*) la *clase* RDF referenciada por la *categoría de entidad* a la que pertenece dicho *atributo*.

En el ejemplo, un *atributo* especificado en el marco C-INCAMI referencia la definición de la *propiedad RDF* “*org:writtenInProgLang*” aplicable a (*rdfs:domain*) la *clase RDF* “*org:SoftwareModule*”, referenciado por la *categoría de entidad* a la cual se asocia dicho *atributo*. La definición de la *propiedad RDF* también identifica el tipo de valores que contiene (*rdfs:range*), en este caso, un “*Lenguaje de Programación*” (“*org:ProgrammingLanguage*”) utilizado por los conceptos del módulo *measurement*.

Por otro lado, en el módulo *measurements*, una *Escala Categórica* (*CategoricalScale*) mantiene una referencia (*classURI*) a una *Clase* del vocabulario de RDF Schemas (*rdfs:Class*) que contiene la definición de un concepto del vocabulario RDF de la organización que corresponde a dicha escala. En el ejemplo de la Figura 4.9 una *escala categórica* especificada en el marco C-INCAMI referencia la definición de la *clase RDF* “*org:ProgrammingLanguage*” que identifica concepto del vocabulario de la organización, y es utilizado como el tipo de valor (*rdfs:range*) de la propiedad “*org:writtenInProgLanguage*” referenciada por el *atributo* que cuantifica la métrica que contiene dicha escala categórica. Relacionado al anterior, un *Valor Categórico* (*CategoricalValue*) mantiene una referencia (*instanceURI*) a un *Recurso RDF* (*rdfs:Resource*), identificado en los datos RDF de la organización, que especifica una instancia de alguna *clase RDF* referenciada por la *escala categórica* a la que pertenece dicho *valor categórico*. En el ejemplo, un *valor categórico* especificado en el marco hace referencia al recurso RDF “*org:Java*” cuyo tipo (*rdf:type*) corresponde a la *clase RDF* “*org:ProgrammingLanguage*” referenciada por la *escala categórica* a la cual pertenece dicho *valor categórico*. De la misma forma, una *Unidad* (*Unit*) mantiene una referencia (*classURI*) a una *clase RDF* del vocabulario RDFS que contiene la definición de un concepto del dominio de la organización usado como *unidad* de la *escala numérica* de alguna *métrica*. Aunque no se representa en la Figura 4.9, una *unidad* especificada en el marco C-INCAMI podría referenciar la *clase RDF* “*org:SoftwareModule*” para representar los valores de alguna *métrica* que contabilice módulos de software.

Al igual que los *atributos*, *entidades* y *categorías de entidad* del módulo *requirements*, también los conceptos del módulo *context* pueden ser enlazados a las definiciones del espacio de información (repositorios y otros) del dominio de la organización, particularmente las *propiedades de contexto*, ya que han sido definidas como un tipo especial de *atributo*, y el *contexto* mismo, definido como un tipo especial de *entidad* (recordar la Figura 4.4 en la Sección 4.3.1). Esto hace posible descubrir las entidades –y sus atributos– relevantes a un contexto particular a un proyecto de M&E, a partir de las definiciones de los conceptos y propiedades del vocabulario de la organización así como de los datos correspondientes almacenados en los repositorios de la organización, para conformar las descripciones de dicho contexto. De la misma forma, accediendo a los valores de dichas propiedades en los repositorios de la organización, es posible cuantificar las propiedades de contexto de un proyecto de M&E concreto, facilitando la tarea de captura de la información de contexto.

Medición y Evaluación Sensible al Contexto



Como se introdujo en el Capítulo 1, una de las contribuciones de esta tesis es la aplicación del enfoque sensible al contexto al área de medición y evaluación de calidad en proyectos de software de una organización. En los capítulos 3 y 4 se han presentado los elementos diseñados para el enfoque sensible al contexto a aplicar en dicha área, que incluye una definición y caracterización del término *contexto*, un modelo de información construido sobre conceptos del área de medición y un modelo de proceso que describe las actividades que involucran la captura y procesamiento de la información de contexto relevante, para ser utilizada en el diseño e implementación de la medición y evaluación de calidad de entidades de software. En este capítulo se describen los aspectos particulares del enfoque diseñado respecto del área de aplicación, a saber: se presenta una caracterización de contexto para M&E y se describen los detalles que involucran la especificación, captura y procesamiento de información de contexto que completan la aplicación del enfoque mencionado para esta tesis.

A partir de la caracterización del término contexto presentada en la Sección 3.3.2, se analizan sus componentes a la luz del área de aplicación del enfoque. La entidad de interés, para la cual se especifica y describe el contexto, es la entidad a ser medida y evaluada para satisfacer una necesidad de información específica. Ésta podría ser una instancia de un producto, recurso, proceso, servicio, o cualquier otro activo siendo evaluado. La información de contexto proviene de dicha entidad y de las entidades relacionadas presentes en su contexto. El contexto incluye sólo la información relevante a la situación de la entidad, considerando a la *situación* como la tarea de medición y evaluación de la entidad de interés para satisfacer la necesidad de información enunciada –propósito, concepto foco y requerimientos no funcionales asociados y el punto de vista–, y las entidades relacionadas a la entidad de interés, mientras éstas sean relevantes a la evaluación –es decir, aquellas que pueden afectar la interpretación del concepto foco de la evaluación sobre la entidad correspondiente. Ya que las entidades a evaluar se encuentran involucradas en un proyecto de software implementado por la organización, las entidades relevantes relacionadas a partir de las cuales se extraen las propiedades de contexto pueden incluir (ver Figura 5.1):

- el proyecto en el cual se encuentra involucrada la entidad, incluyendo los recursos, artefactos, productos y otros activos usados en el proyecto que afectan a la entidad de alguna forma;
- los procesos aplicados a la entidad;
- la organización que lleva a cabo el proyecto;
- las personas o agentes interesados en los resultados de la evaluación, pudiendo estos pertenecer o no a la organización;

- el ambiente o entorno externo a la organización.

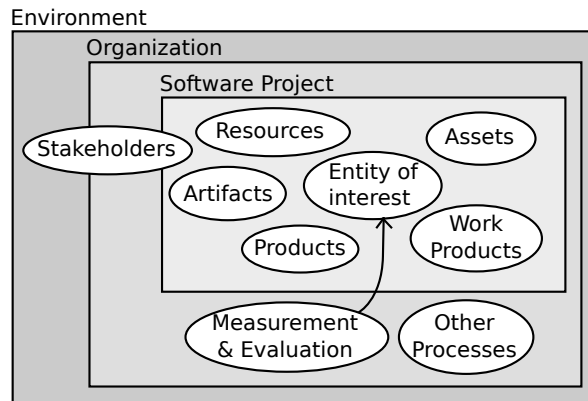


Figura 5.1: Componentes de la situación relativa a una entidad de interés sometida a medición y evaluación a partir de la cual se extrae la información de contexto relevante.

Utilizando el mecanismo de integración de las especificaciones realizadas mediante el marco C-INCAMI al espacio de información de la organización, presentado en el capítulo anterior, es posible obtener la información de las propiedades de contexto, y las entidades a las que pertenecen, de los repositorios de la organización, así como los valores de las propiedades en cada entidad, utilizados para describir un contexto. Recuerde que los repositorios de la organización contienen el modelo o esquema de dominio de la organización así como los datos que conforman con tales esquemas. De esta forma, mediante la exploración o navegación del espacio de información de la organización, es posible descubrir las entidades relacionadas a la entidad de interés, así como las propiedades de éstas, que pudieran ser relevantes a la situación, para incorporarlas a la descripción del contexto del proyecto de medición y evaluación. Asimismo, los metadatos y datos de propiedades de contexto reusables y descripciones de contexto particulares pueden ser almacenados en repositorios de la organización para ser reutilizados en proyectos futuros, favoreciendo de esta forma la consistencia entre las descripciones de contexto realizadas para diferentes proyectos a lo largo del tiempo que posibilite una comparación más robusta e interpretación más coherente de los resultados de proyectos de medición y evaluación. Si la organización no dispusiera de estos repositorios, la adopción del marco C-INCAMI podría ser utilizada como punto de partida para la incorporación de los mismos como parte integrada de los sistemas de información y gestión de la organización (como también se comentó en la Sección 4.6).

Del análisis previo, podemos resumir una definición de *contexto* adaptada a M&E como *una entidad especial, descrita por un conjunto de propiedades de contexto que caracterizan el estado relevante de la situación de una entidad sujeta a M&E. La situación de la entidad está definida por la necesidad de información y su interacción con otras entidades –a partir de las cuales se obtienen las propiedades de contexto–, respecto de dicha necesidad, mientras ellas afecten la implementación e interpretación de M&E para la entidad objeto.*

5.1. Especificación de Contexto y Contextualización

La aplicación del enfoque *sensible al contexto* requiere, como se vio en el Capítulo 3, diseñar un modelo que permita representar información de contexto así como un modelo de proceso que describa cómo capturar y utilizar la misma para el propósito de la aplicación. En la subsección 4.3.1 se presentaron los componentes y relaciones del modelo de contexto, incluidos en el módulo *context* del marco C-INCAMI, que permiten especificar las propiedades utilizadas para describir un contexto e inclusive los criterios utilizados por las mismas para realizar comparaciones de similitud. Sin embargo, los detalles

sobre cómo se lleva a cabo la cuantificación de dichas propiedades fue pospuesto hasta el presente capítulo.

Tal como se presentó en la Sección 3.3.3, las propiedades de contexto pueden ser cuantificadas aplicando el mecanismo de medición de atributos del marco INCAMI. Esto es posible ya que el modelo de contexto diseñado extiende el comportamiento de los conceptos del marco utilizados para tal propósito. En la Figura 5.2 se muestra el modelo de contexto presentado en el capítulo anterior, agregando las relaciones con los conceptos del módulo *measurement* (presentado en la Sección 4.4 e ilustrado en la Figura 4.6) que permiten hacer uso de tales conceptos de forma consistente en la descripción de contextos.

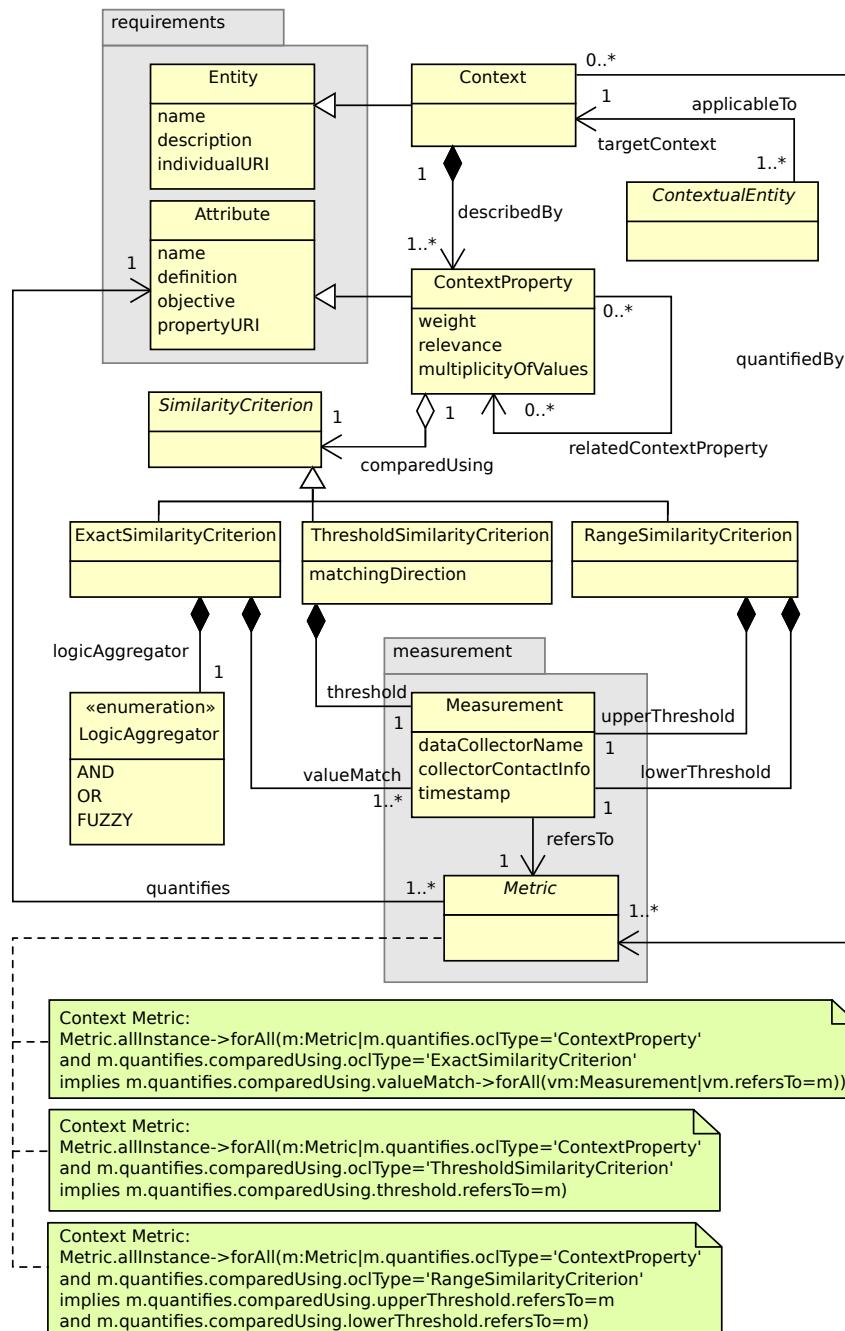


Figura 5.2: Diagrama de clases (UML) mostrando todos los conceptos y relaciones del módulo *context* del marco C-INCAMI.

De esta forma, las propiedades de contexto corresponden a atributos asociados a alguna categoría de entidad a la cual pertenecen las entidades relevantes involucradas en el contexto (como se verá más adelante en este capítulo, no siempre se dispone de las entidades concretas involucradas en una descripción de contexto). Para cada propiedad de contexto se selecciona/especifica una métrica que define cómo se obtienen y representan los valores para dicha propiedad. Cada métrica que cuantifica una propiedad de contexto es asociada al (objeto) *contexto* correspondiente (en la medición con fines de evaluación, como se describió en la Sección 4.4, las métricas son asociadas al *Proyecto de Medición*). De esta forma, utilizando la métrica asignada a cada propiedad, se efectúan mediciones y los resultados (objetos *Measurement*) son asociados a la entidad sobre la cual se realizó la medición o sobre el (objeto) *contexto* mismo –dependiendo de la situación para la cual se describa un contexto, como se verá luego.

Por otro lado, en la sección 4.3.1 se describieron los *criterios de similitud* (*SimilarityCriterion*) que pueden utilizarse para comparar los valores de una misma propiedad en diferentes descripciones de contextos. Estos criterios especifican un conjunto de parámetros –tales como *threshold*, *upperThreshold*, *lowerThreshold* y *valueMatch*– que son utilizados en cada caso para determinar un valor de similitud. Ya que estos parámetros corresponden a valores que son asignados a las propiedades de contexto, y ya que dichos valores son representados mediante mediciones (y sus medidas asociadas), en el modelo extendido de la Figura 5.2 se especifica que tales parámetros hacen referencia a las mediciones (*Measurement*) obtenidas a partir de la métrica que cuantifica la propiedad de contexto que posee el criterio de similitud en cuestión (tal como lo indica la restricción OCL¹ asociada a tales relaciones). De esta forma, se mantiene la consistencia de tipos de valor (especificados en la métrica correspondiente) entre los parámetros de los criterios y de los valores que pudieran ser asignados a la propiedad mediante la cuantificación de la entidad correspondiente, asegurando así la coherencia de las comparaciones realizadas.

Para alcanzar los beneficios de la aplicación del enfoque utilizando los elementos descriptos hasta aquí se debe recurrir, a nivel del modelo de información, a la asociación de las descripciones de contexto con los elementos de información del área de aplicación correspondiente. A esta asociación de descripciones de contexto a la información de dominio se le denomina *contextualización*. La contextualización de información de dominio es la base para la documentación de experiencias y la implementación de mecanismos sensibles al contexto. En el Capítulo 3 se describió el enfoque de modelado de contexto aplicado a este trabajo y, utilizando dicho enfoque, se presentó un diseño, introducido primero en el Capítulo 4 y luego descrito en detalle aquí, para representar descripciones de contexto. A partir de estos modelos se pueden distinguir dos escenarios de contextualización de la información relacionada a M&E. Primero, a la *necesidad de información* de un proyecto de M&E se asocia un *contexto* para describir el *contexto real* que caracteriza la situación actual de la entidad siendo evaluada en dicho proyecto (en la Figura 4.3, Sección 4.3, el concepto *InformationNeed* mantiene una relación *characterizedBy* con un objeto *Context*). Segundo, a una *entidad contextual* se asocia un *contexto* para describir el *contexto de aplicación* que caracteriza la situación ideal en la que dicha entidad, por su definición, puede ser aplicada y/o interpretada coherentemente (en la Figura 5.2 el concepto *ContextualEntity* mantiene una relación *applicableTo* con un objeto *Context*). Como también se comentó en el Capítulo 4, los conceptos del dominio de aplicación que son sensibles al contexto se especifican extendiendo o especializando el concepto *ContextualEntity*. Allí se vio que los conceptos *ConceptModel*, *Attribute*, *Metric*, *ElementaryModel* y *GlobalModel* son los conceptos del dominio de M&E del marco C-INCAMI especificados como entidades contextuales, es decir, para los cuales es posible especificar un *contexto de aplicación* (ver figuras 4.5, 4.6 y 4.7 del Capítulo 4).

En cualquiera de los escenarios de contextualización identificados, la especificación del contexto requiere de dos pasos generales: primero, se identifican las propiedades (*ContextProperty*) que describirán el contexto relevante a la situación correspondiente y se asignan al contexto respectivo (*Context*);

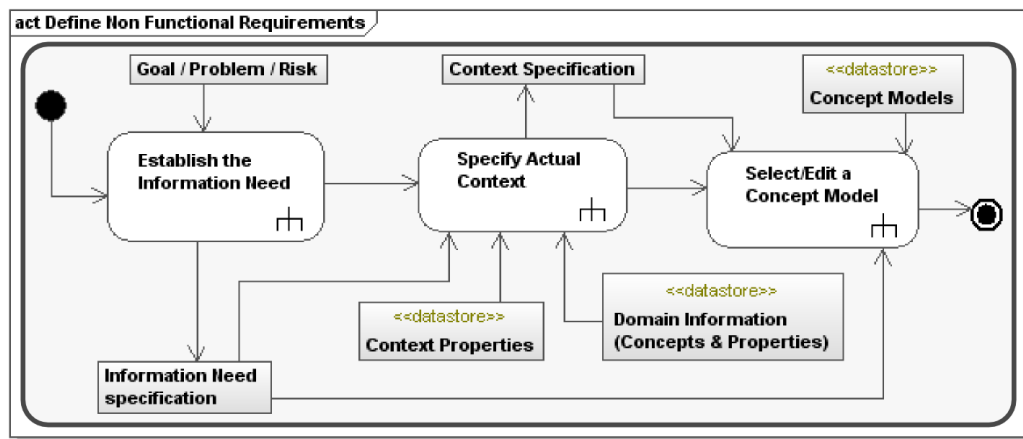
¹*Object Constraint Language* - Lenguaje declarativo para describir restricciones a las relaciones en los modelos UML (*Unified Modeling Language*).

segundo, las propiedades de contexto identificadas son cuantificadas –aplicando los conceptos del módulo *measurements*, como se introdujo en el Capítulo 3- para caracterizar al contexto correspondiente. Estos dos escenarios de contextualización son utilizados, como se verá en la sección 5.2, como base en la definición de los mecanismos para implementar la funcionalidad sensible al contexto para el dominio de medición y evaluación.

A continuación se discuten las particularidades en la especificación de contexto para los dos escenarios de contextualización presentados.

5.1.1. Contexto Real

El *contexto real* que caracteriza la situación actual, definida por la necesidad de información en un proyecto de M&E, se describe durante la definición de los requerimientos no funcionales (recordar el proceso de M&E ilustrado en la Figura 3.3, Sección 3.2.2), luego de establecer la necesidad de información del proyecto y previamente a seleccionar y editar el modelo de conceptos que define de forma explícita los requerimientos no funcionales (ver Figura 5.3). La especificación detallada del proceso completo de M&E para el marco C-INCAMI está fuera del alcance del propósito de esta tesis².



Generated by UModel

www.altova.com

Figura 5.3: Actividades involucradas en la especificación de los requerimientos no funcionales, incorporando la especificación del *contexto real*.

El contexto real se especifica, tal como se ilustra en la Figura 5.4, identificando las entidades relevantes involucradas en el mismo y sus propiedades –también relevantes- que serán cuantificadas para caracterizar de forma efectiva el contexto. Las entidades relevantes involucradas en el contexto real pueden ser descubiertas, mediante el mecanismo de integración –presentado en la Sección 4.6-, inspeccionando en los catálogos de la organización las relaciones de la entidad a ser evaluada con otras entidades –pertenecientes a diferentes categorías tales como productos, procesos, proyectos, recursos, agentes y la organización misma (recordar la Figura 5.1). Una vez seleccionadas las entidades relevantes del contexto real se seleccionan, nuevamente de los catálogos de la organización, las propiedades relevantes asociadas a las mismas para ser incluidas a la descripción del contexto. Las propiedades de contexto pueden obtenerse directamente del *catálogo de contexto* o ser especificadas a partir de especificaciones de propiedades almacenadas en los catálogos de dominio de la organización (como se indicó en la sección 3.3.3). Las propiedades del contexto relevantes pueden ser descubiertas explorando, en el catálogo de contexto, las propiedades asociadas con la entidad evaluada, o entidades relacionadas a ésta, o contemplando las propiedades de contexto relacionadas a aquellas propiedades que ya fueron

²Al momento de escritura de esta tesis, dicha especificación está siendo desarrollada en el marco de la investigación relacionada al marco C-INCAMI, como objeto de otra tesis doctoral.

incluidas en el contexto real (recordar la relación *relatedContextProperty* de la clase *ContextProperty*, descrita en la Sección 4.3.1). La elección de incluir una propiedad como relevante para describir el contexto real es una decisión del evaluador y su fundamento debe ser registrado para cada propiedad de contexto (ver atributo *relevance* del concepto *ContextProperty*, descrito en la Sección 4.3.1). Asimismo, se debe especificar el *peso* (*weight*) que representa la importancia relativa de cada propiedad agregada al contexto, teniendo en cuenta que la suma de los pesos de todas las propiedades del contexto debe dar 1. También se debe indicar para cada propiedad la cantidad de valores (mediciones) que pueden ser asignados a las mismas (*multiplicityOfValues*) pudiendo indicar un número entero positivo o el valor 0, en cuyo caso se puede asignar a la propiedad una cantidad de valores arbitraria. Finalmente, resta asignar a cada propiedad un criterio de similitud –en el caso del contexto real se asigna por defecto un *Criterio de Similitud Exacto* (*ExactSimilarityCriterion*) con un operador lógico de agregación FUZZY (en la Subsección 5.2.1 se aclara esta decisión)-, en el cual cada uno de los valores que componen el criterio (*valueMatch*) serán tomados de las mediciones asignadas a la propiedad, como se describe a continuación.

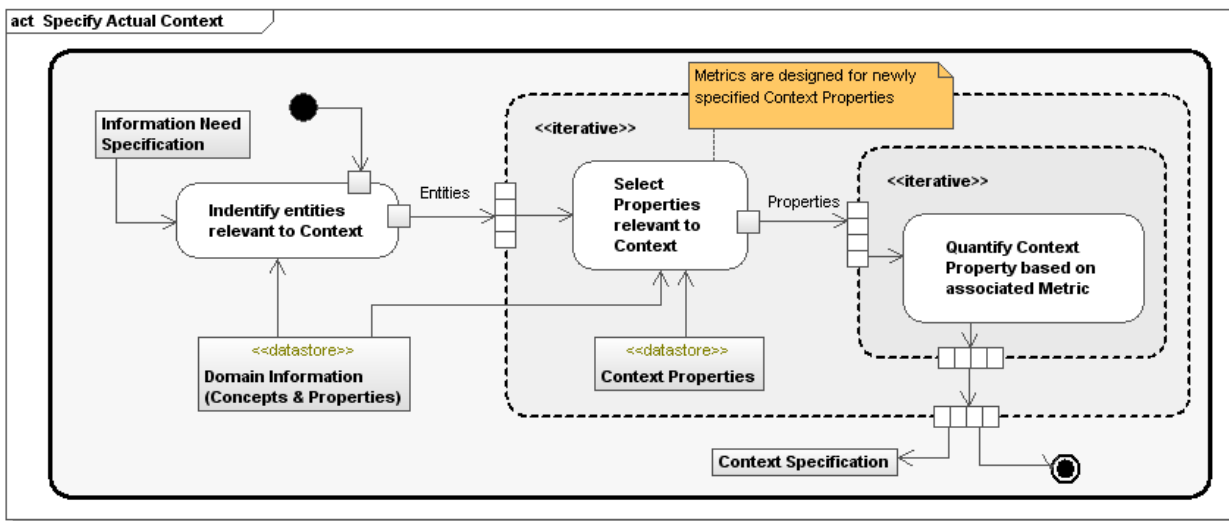


Figura 5.4: Actividades involucradas en la especificación del *contexto real*, durante la especificación de requerimientos no funcionales.

Para completar la descripción del contexto real, las propiedades de las entidades involucradas en el contexto, identificadas como relevantes, deben ser cuantificadas. Como se describió al comienzo de la Sección 5.1, el modelo de contexto se construye sobre los conceptos y relaciones del módulo *measurement* del marco INCAMI; particularmente, una propiedad de contexto es considerada una especialización de un *Atributo* (recordar la Figura 5.2). De esta forma es posible asociar una métrica a cada propiedad de contexto y, siguiendo su especificación, realizar la medición sobre la entidad asociada a la propiedad, incluida en el contexto real, para obtener un valor para la misma que caracterice de forma efectiva el contexto. Las métricas pueden ser seleccionadas de un repositorio que contenga especificaciones de métricas para los atributos de las entidades del dominio de aplicación, si se dispone de tal, o pueden ser diseñadas, en cuyo caso se almacenan en el mismo. Las métricas seleccionadas para cuantificar las propiedades del contexto son asociadas al mismo *contexto*. Al igual que en el caso de las mediciones de atributos con fines de evaluación, más de una medición puede ser realizada para una propiedad de contexto (ver Sección 4.4), siempre y cuando así lo indique el valor del atributo *multiplicityOfValues* de la propiedad (como se mencionó arriba). Cada una de las mediciones registradas para una propiedad de contexto serán agregadas a los valores (*valueMatch*) que componen el *criterio de similitud exacto* asignado a la misma.

Para clarificar el escenario de la especificación de un *contexto real*, considere el ejemplo ilustrado en

la Figura 5.5 en el que se describe el *contexto real* que caracteriza la necesidad de información especificada para evaluar el concepto de “Calidad de Contenido” del “Carro de compras de Cuspide”³. En este contexto se encuentra involucrada la entidad relevante “Cuspide” –una *aplicación web de e-commerce* (categoría de entidad “eComWebApp”)– de la cual la propiedad “lenguaje natural soportado” es relevante para describir el contexto (entre otras, según se refleja en el peso indicado para la misma). Esta propiedad es relevante ya que se está evaluando el concepto “Calidad de Contenido” –el lenguaje en que está escrito el contenido puede determinar qué métricas pueden usarse y cómo pueden interpretarse los valores medidos (indicadores utilizados). Esta propiedad es cuantificada por una métrica cuya escala categórica permite especificar valores que identifican diferentes lenguajes naturales, tales como “español”, “inglés”, “portugués”, entre otros. Las mediciones realizadas sobre la entidad “Cuspide” para dicha propiedad de contexto, utilizando la métrica mencionada, producen como resultado los dos lenguajes naturales soportados por la aplicación web: “español” e “inglés”⁴. A la propiedad de contexto se asigna, como se indicó arriba, un *criterio de similitud exacto*, con un operador lógico de agregación FUZZY, incluyendo las dos mediciones obtenidas para la propiedad como los valores (*valueMatch*) incluidos en el criterio.

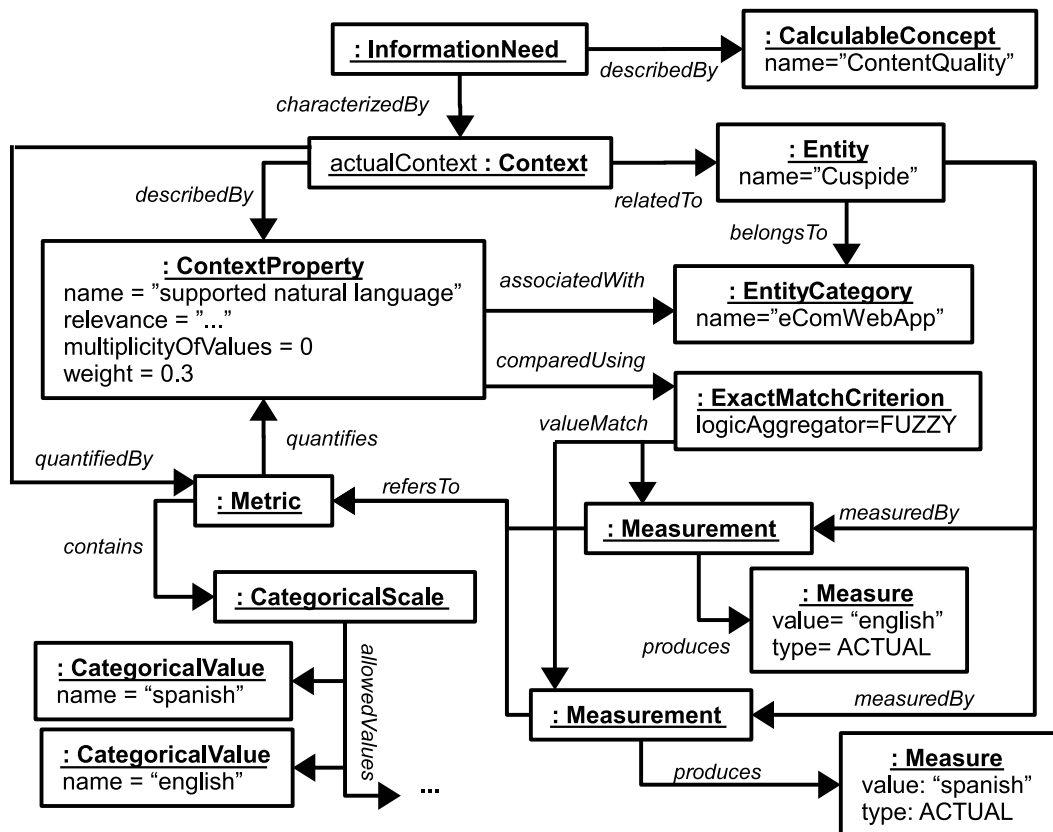


Figura 5.5: Diagrama de objetos (UML) describiendo un ejemplo de la especificación de un *contexto real*.

5.1.2. Contexto de Aplicación

El *contexto de aplicación*, que caracteriza la situación ideal en la que una *entidad contextual* es aplicable, se especifica *a priori* e independientemente de cualquier proyecto de M&E particular y debe ser almacenado junto a la entidad contextual correspondiente en el catálogo de la organización. Al igual que en el contexto real, la descripción de un contexto de aplicación incluye un conjunto de

³<http://www.cuspide.com>

⁴Estos valores son ficticios. A la fecha, *Cuspide.com* no soporta lenguajes extranjeros.

propiedades de contexto. Sin embargo, y a diferencia del primero, en un contexto de aplicación no existen entidades concretas involucradas ya que los valores que lo describen son valores estimados o esperados, y no reales. Así, las propiedades de contexto relevantes a un contexto de aplicación pueden ser descubiertas explorando, en el espacio del dominio de aplicación de la organización, las categorías de entidad relacionadas directa o indirectamente a la entidad contextual. Por ejemplo, según se vio en la Sección 4.4, una *Métrica* es una entidad contextual (recordar la Figura 4.6); entonces, para describir el contexto de aplicación de una determinada métrica se puede navegar hacia el *atributo* que cuantifica la misma (relación *quantifies*) y luego hasta la *categoría de entidad* a la que pertenece dicho atributo (relación *associatedWith*), y de aquí a otras categorías de entidad relacionadas (relación *relatedEntityCategory*). Al igual que en el contexto real, debe especificarse para cada propiedad de contexto la razón por la cual es incluida en la descripción de dicho contexto (atributo *relevance*), así como su peso y la cantidad de valores (mediciones) que pueden especificarse para la misma. También se debe asignar un criterio de similitud (*SimilarityCriterion*) apropiado para reflejar los valores de la propiedad que describen el contexto al cual es aplicable la entidad contextual, pudiendo elegir cualquiera de los tres criterios vistos en la Subsección 4.3.1 (aunque siguiendo los lineamientos presentados en 5.2.1.3).

También en un *contexto de aplicación* las propiedades son cuantificadas mediante mediciones a partir de las métricas que se asignan a las mismas. No obstante, como no existen entidades reales involucradas en el contexto de aplicación, las mediciones corresponden a valores estimados o esperados (las *medidas* son de tipo *ESTIMATED*, según se representa en la Figura 4.6, Sección 4.4) y se asocian al *contexto* mismo mediante la relación *measuredBy* (recordar que en la Figura 5.2 *Context* es una especialización de *Entity*). Al igual que en el *contexto real*, múltiples mediciones pueden registrarse para una propiedad de un contexto de aplicación, y las mismas deben ser asignadas a los valores correspondientes en la especificación del criterio de similitud asignado. No obstante, a diferencia del contexto real, un contexto de aplicación describe el contexto ideal en que una entidad contextual es aplicable; y ya que sostenemos que una entidad contextual puede ser aplicable a más de un contexto real, los valores asignados a cada propiedad, junto al criterio de similitud seleccionado, representan los diferentes valores que estos diferentes contextos pueden presentar para dicha propiedad.

Consideremos ahora un ejemplo para aclarar este escenario, ilustrado en la Figura 5.6. La métrica "*Lexile text*"⁵ que cuantifica el atributo "*Legibilidad de Texto*", asociado a la categoría de entidad "*Página Web*", es considerada una entidad contextual para la cual se describe su *contexto de aplicación* (*targetContext*). Éste se describe únicamente mediante la propiedad de contexto "*lenguaje natural soportado*", también asociada a una "*Página Web*". Esta propiedad es considerada relevante en este contexto ya que la aplicabilidad de la métrica depende del lenguaje natural en el que está escrito el texto sobre el que se aplique –siendo útil solo para los lenguajes "*inglés*" o "*español*". También se indica, asignando el valor "0" al atributo *multiplicityOfValues*, que un número arbitrario de mediciones son aceptables para la propiedad. Esta propiedad es cuantificada, como en el ejemplo visto en el caso de *contexto real* –en la subsección anterior-, por una métrica cuya *escala categórica* especifica valores correspondientes a diferentes lenguajes naturales ("*español*", "*inglés*", etc.). Por lo tanto, para la propiedad de contexto "*lenguaje natural soportado*" se registran dos mediciones con sus medidas correspondientes (indicando en cada caso que se tratan de estimaciones), asociándolas a un *criterio de similitud exacto*, con un operador de agregación lógica *OR*, para describir los dos posibles valores (*valueMatch*) que describen el contexto al cual es aplicable la métrica "*Lexile text*". Notar que, a diferencia del caso del contexto real, las mediciones para las propiedades de contexto son asociadas al mismo (objeto) *contexto*.

Cabe notar, a partir de las figuras de ejemplo de ambos escenarios de contextualización vistos, que el modelo de contexto utiliza en gran medida los conceptos y relaciones de la base conceptual (el

⁵<http://www.lexile.com/about-lexile/lexile-overview/>

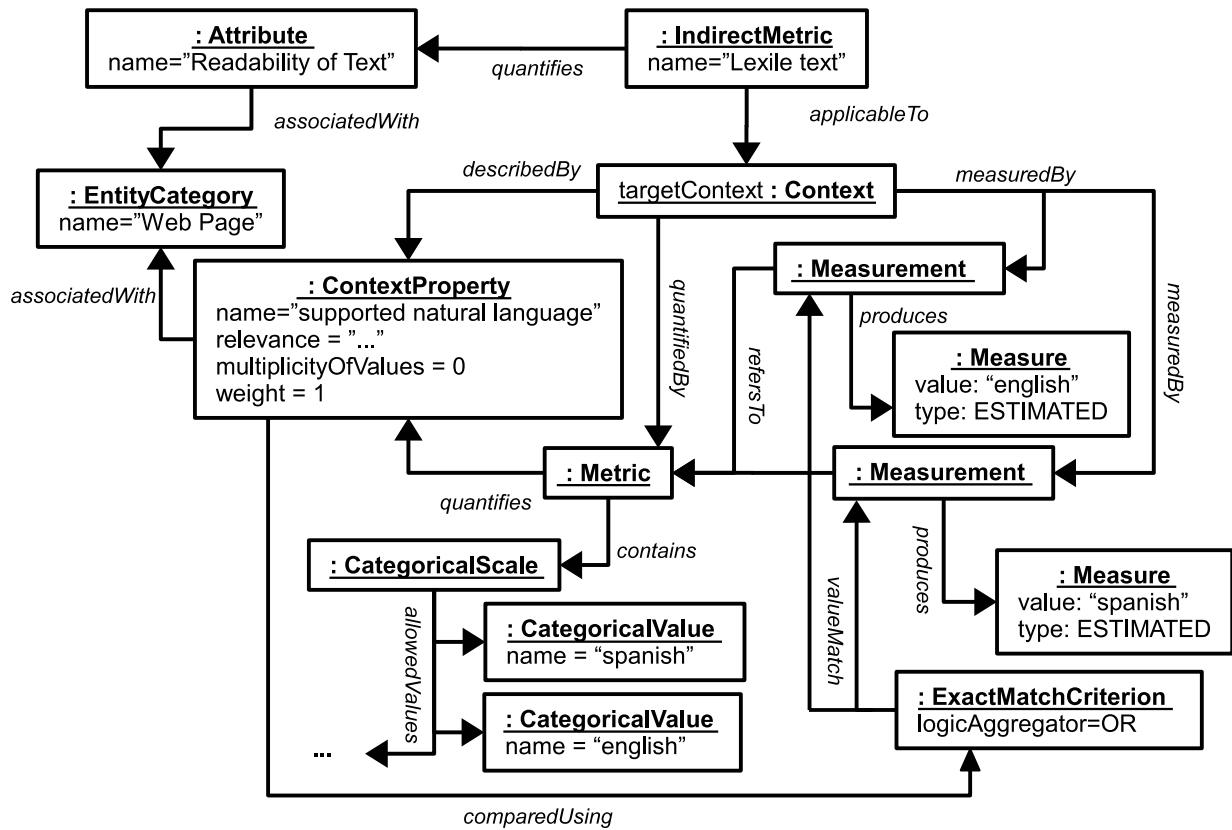


Figura 5.6: Diagrama de objetos (UML) describiendo un ejemplo de la especificación de un *contexto de aplicación*.

módulo *measurement* de *INCAMI*) reduciendo la carga adicional de términos necesarios para describir información de contexto.

5.2. Uso de Información de contexto

En la sección anterior se describieron los aspectos estructurales y procedurales que permiten llevar a cabo la captura y especificación de información de contexto para los dos escenarios de contextualización identificados en el dominio de M&E sensible al contexto tratado en esta tesis. Para hacer efectiva la utilidad de la contextualización de información de M&E, como se indicó en la Sección 3.3, se requieren mecanismos para procesar la información de contexto especificada, asociada a la información de dominio correspondiente, para ser utilizada en la implementación de la funcionalidad sensible al contexto, objetivo de la aplicación del enfoque en este trabajo.

Como se adelantó en la Subsección 1.2.1, en este trabajo la información de contexto es usada en los siguientes escenarios de M&E de calidad:

- *Interpretación de resultados de proyectos de M&E:* la necesidad de información enunciada para cada proyecto de M&E incluye la descripción del *contexto real*, relevante a la entidad evaluada y en el que fue desarrollada la evaluación. Esta información forma parte de la documentación del proyecto y es almacenada junto con los diseños de M&E y resultados del mismo, por lo que puede ser consultada para ayudar a interpretar coherentemente los resultados arrojados por la evaluación.
- *Comparación de resultados entre diferentes proyectos de M&E de la organización:* la comparación de resultados entre diferentes proyectos de M&E debe realizarse sólo cuando los mismos hayan

sido llevados a cabo en contextos similares –asumiendo que ambos son comparables sobre la base de la categoría de entidad, los requerimientos no funcionales, métricas y otros-, de otra forma, los resultados del análisis pueden no ser confiables. Las descripciones del *contexto real* incluidas en la especificación de la necesidad de información de cada proyecto de M&E pueden ser comparadas para determinar esta similitud.

- *Diseño de las actividades de M&E:* como se vio en las secciones 4.4 y 4.5, el diseño de estas actividades consiste básicamente en la selección de elementos de información reusables –métricas e indicadores- almacenados en repositorios de la organización, disponibles a todos los proyectos de la misma. Para proveer la funcionalidad de selección sensible al contexto, el *contexto de aplicación* asociado a estos elementos reusables, sensibles al contexto –por ejemplo, una métrica para cuantificar un atributo dado- puede ser comparado con el *contexto real* de un proyecto de M&E para determinar su aplicabilidad o compatibilidad con el mismo.

En la Figura 5.7 se ilustra el proceso de M&E, originalmente presentado en la Sección 3.2.2, incorporando los elementos propuestos por el enfoque *sensible al contexto* aplicado en el marco C-INCAMI que dan soporte a los escenarios mencionados. Básicamente, estos elementos corresponden a: *i)* un catálogo organizacional de *Propiedades de Contexto*, *ii)* un repositorio organizacional de información de dominio y *iii)* la especificación de contexto de cada Proyecto de M&E (como parte de la especificación de requerimientos no funcionales). Estos elementos son utilizados como entradas a las actividades correspondientes: *Definir los Requerimientos No Funcionales*, *Diseñar la Medición*, *Diseñar la Evaluación* y *Analizar y Recomendar*.

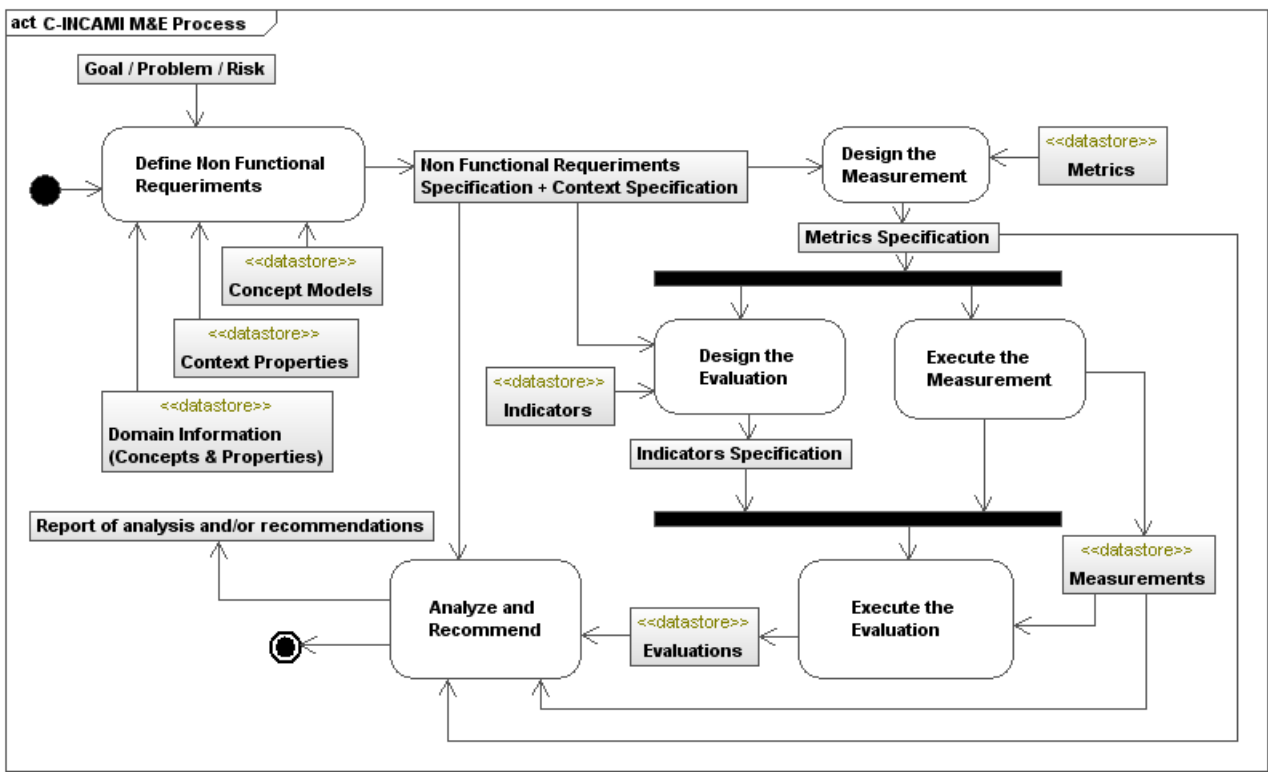


Figura 5.7: Proceso de M&E incorporando los elementos del enfoque *sensible al contexto*.

Para satisfacer estos escenarios, particularmente los dos últimos, se ha diseñado un mecanismo de comparación de descripciones de contexto que produce un valor que expresa de forma objetiva, en el primer caso, el grado de similitud de dos *contextos reales* dados y, en el segundo caso, el grado en que un entidad contextual particular es aplicable a un *contexto real* dado. Este mecanismo se describe a continuación.

5.2.1. Comparación de Contexto

Arriba se describieron los escenarios en los que la información de contexto es usada en el marco de M&E C-INCAMI, enfatizando su uso en el diseño de mecanismos para automatizar recomendaciones durante el diseño de actividades de M&E así como durante el análisis comparativo entre resultados de proyectos de M&E. Tal como se mencionó, este mecanismo se basa en la comparación de similitud de descripciones de contextos correspondientes a los escenarios descriptos, pudiendo ser utilizada para determinar si una entidad contextual puede ser coherentemente aplicada en un proyecto de M&E dado, por ejemplo cuando se selecciona una métrica durante el diseño de medición, o si los resultados de dos proyectos de M&E pueden realmente ser comparados para tomar las decisiones apropiadas [Molina *et al.*, 2010]. El mecanismo diseñado para llevar a cabo estas comparaciones se basa en el uso de un conjunto de métricas aplicables a los dos escenarios mencionados. Las dos métricas principales del mecanismo de comparación son las siguientes:

- *Similitud Contextual*: permite comparar dos *contextos reales* provenientes de diferentes proyectos de M&E para determinar el grado de similitud contextual de dichos proyectos, y
- *Aplicabilidad de Entidad Contextual*: permite comparar el *contexto de aplicación* de una entidad contextual contra el *contexto real* de un proyecto de M&E para determinar el grado o nivel de aplicabilidad contextual de dicha entidad al proyecto.

Los valores de ambas métricas deben ser interpretados como más similar/aplicable cuanto más cerca estén de 1.

Como se puede prever, estas métricas utilizan los criterios de similitud asociados a las propiedades de contexto (como se describió en la Subsección 4.3.1) para establecer, tanto la similitud entre dos contextos reales como la aplicabilidad entre un contexto de aplicación y uno real. Cada criterio de similitud se traduce en una fórmula o ecuación que calcula, dada una propiedad de contexto y dos contextos, la similitud entre los valores especificados para dicha propiedad en sendos contextos.

A continuación se describen las métricas diseñadas para implementar las comparaciones mencionadas, así como las ecuaciones asociadas a cada criterio de similitud (incluyendo sus variantes de acuerdo a los parámetros que pueden ser indicados en cada caso). Para facilitar la comprensión de las mismas, se utilizará una notación matemática semiformal para describir cada uno de sus componentes. No obstante, las fórmulas o ecuaciones presentadas constituyen una representación de la especificación incluida en dichas métricas, tal como se definió en la Sección 4.4. Antes de entrar en detalles de cada métrica, presentamos la notación que se usará:

- Un *contexto* se describe mediante una tripla (CP, CV, CW) donde:
 - $CP = \{cp_1, \dots, cp_n\}$ es el conjunto de propiedades de contexto especificadas para un contexto dado; ellas representan únicamente sus metadatos (la definición de la propiedad), tal como se encuentran especificadas y almacenados en el repositorio de definiciones del dominio de aplicación de la organización,
 - $CV = \{CVcp_1, \dots, CVcp_n\}$ son los conjuntos de valores que describen un contexto, un conjunto por cada propiedad de contexto, donde $CVcp_i = \{cv_1, \dots, cv_m\}$ es el conjunto de valores especificados para la propiedad de contexto cp_i ;
 - $CW = \{CWcp_1, \dots, CWcp_n\}$ es el conjunto de pesos relativos para cada propiedad de contexto cp_i en el contexto dado; donde cada $CWcp_i \in \mathbb{R}$ y $\sum_{i=1}^n CWcp_i = 1$.
- Particularmente, denotamos con *AC* (*Actual Context*) a un *contexto real* y con *TC* (*Target Context*) a un contexto de aplicación, de forma que:

- $AC(a) = (CP(a), CV(a), CW(a))$ describe el *contexto real* de un proyecto a , y
- $TC(e) = (CP(e), CV(e), CW(e))$ describe el *contexto de aplicación* de una entidad contextual e .

También cabe recordar, como se indicó en la Subsección 4.3.1, que llamamos *equivalentes* a dos propiedades de contexto incluidas en sendas descripciones de contexto pero proveniente de una misma especificación extraída del repositorio de dominio de la organización, es decir, se trata de la misma propiedad de contexto incluida en la descripción de dos contextos diferentes.

5.2.1.1. Similitud Contextual (*CSim*)

Esta métrica se aplica en la comparación de dos *contextos reales* provenientes de diferentes proyectos para determinar su similitud contextual y establecer si pueden ser comparados ambos proyectos de forma coherente. El valor producido por la métrica *CSim* representa el porcentaje o proporción de propiedades en un contexto real de un proyecto de M&E que son similares a las propiedades equivalentes de un segundo contexto real. La similitud entre propiedades de contexto equivalentes de ambos contextos se determina usando la métrica *CPSim* (presentada luego en la Sección 5.2.1.3).

Así, dados los contextos reales $AC(a)$ y $AC(b)$ (tal como se indicó arriba) especificados para los proyectos a y b respectivamente, definimos:

- $CP(a, b) = CP(a) \cup CP(b)$ como las definiciones de todas las propiedades de contexto diferentes especificadas para ambos contextos $AC(a)$ y $AC(b)$ en conjunto, y
- $|CP(a, b)|$ como la cantidad de propiedades en $CP(a, b)$

Entonces, la *similitud contextual CSim* es calculada como la suma de la similitud entre las propiedades equivalentes de $AC(a)$ y $AC(b)$ para todas las propiedades incluidas en $CP(a, b)$, ponderada por el peso de cada propiedad cp_i :

$$CSim(AC(a), AC(b)) = \sum_{i=1}^{|CP(a,b)|} CWcp_i(a) \times CPSim(cp_i, AC(a), AC(b)) \quad (5.1)$$

Donde $CWcp_i(a)$ es el peso de la propiedad de contexto cp_i en el contexto que caracteriza la entidad a (en la Ecuación 5.1 el peso de cada cp_i es tomado del contexto real $AC(a)$) y *CPSim* representa el cálculo de similitud entre las propiedades equivalentes cp_i especificadas en ambos contextos $AC(a)$ y $AC(b)$.

Se debe notar que, como lo indica el límite de la sumatoria de la ecuación, todas las propiedades especificadas en ambos contextos son consideradas en conjunto como base para realizar la comparación, sin importar si están presentes en ambos contextos al mismo tiempo, ya que se trata de una comparación entre dos contextos del mismo tipo (ambos son *contextos reales*).

La especificación de la métrica *CSim*, siguiendo las definiciones del marco C-INCAMI, se presenta en la Tabla 5.1.

5.2.1.2. Aplicabilidad de Entidad Contextual (*CApp*)

La métrica presentada a continuación puede ser usada para determinar si una entidad contextual dada puede ser aplicada coherentemente a un proyecto de M&E dado. La métrica compara el *contexto de aplicación* de una entidad contextual con el *contexto real* de un proyecto para determinar el grado de aplicabilidad de la entidad contextual al proyecto. En este caso sólo las propiedades especificadas para el contexto de aplicación son tomadas como base para la comparación, a diferencia de lo que ocurre en la métrica *CSim* (Ecuación 5.1) donde todas las propiedades especificadas para ambos contextos

Tabla 5.1: Especificación de la métrica “Similitud Contextual” (*CSim*) usando la definiciones provistas por el marco C-INCAMI.

INDIRECT METRIC			
Name	Similitud Contextual (<i>CSim</i>)	Objective	Determinar la similitud entre dos descripciones de contexto.
References	—	Accuracy	100 %
Author	Hernán Molina	Version	1.0
Value interpretation	Grado o proporción en la que dos descripciones de contexto son similares.		
Related Metric	Similitud de Propiedad de Contexto (<i>CPSim</i>)		
SCALE (NUMERICAL SCALE)			
Value type	REAL	Scale type	RATIO
Representation	CONTINUOUS		
Range Min Value	0.0	Name	Similitud de Contexto
Range Max Value	1.0	Description	Grado o nivel de similitud entre dos descripciones de contexto.
Step Value	—	Acronym	sc
FUNCTION			
Name	Fórmula <i>CSim</i>		
Specification	$CSim(AC(a), AC(b)) = \sum_{i=1}^{ CP(a,b) } CWcp_i(a) \times CPSim(cp_i, AC(a), AC(b))$		
CALCULATION METHOD			
Name	Cálculo de <i>CSim</i>		
Specification	Calcular la Similitud de Propiedad de Contexto (<i>CPSim</i>) para cada propiedad del contexto de referencia respecto de la propiedad equivalente del contexto comparado y luego calcular la suma de las similitudes ponderadas por el peso de la propiedad respectiva en el contexto de referencia.		
References	—		

en conjunto eran tomadas como base para la comparación. Ahora, la comparación es realizada sobre la base del *contexto de aplicación* en lugar del *contexto real*, ya que éste último puede especificar propiedades no relevantes a la entidad contextual. El valor producido por la métrica *CApp* representa el porcentaje o proporción de propiedades en el contexto de aplicación de una entidad contextual que son similares a las propiedades equivalentes de un contexto real. La similitud entre propiedades de contexto equivalentes en diferentes contextos se determina usando la métrica *CPSim* (presentada más abajo).

Por lo tanto, dado el *contexto real* $AC(p)$ (tal como se indicó anteriormente) especificado para el proyecto p y el *contexto de aplicación* $TC(e)$ especificado para la entidad contextual e , la aplicabilidad contextual *CApp* es calculada como la suma de la similitud entre las propiedades equivalentes de $TC(e)$ y $AC(p)$ para todas las propiedades incluidas en $CP(e)$, ponderada por el peso de cada propiedad cp_i :

$$CApp(TC(e), AC(p)) = \sum_{i=1}^{|CP(e)|} CWcp_k(e) \times CPSim(cp_i, TC(e), AC(p)) \quad (5.2)$$

Donde $CWcp_k(e)$ es, como en la métrica anterior, el peso de la propiedad indicada en el contexto de la entidad indicada, y *CPSim* es la métrica que representa el cálculo de similitud entre los valores de la propiedad cp_i especificada en ambos contextos, el *contexto de aplicación* de la entidad e , $TC(e)$, y en el *contexto real* del proyecto p , $AC(p)$, que será descripta luego.

Vale mencionar que para las métricas discutidas arriba, la cantidad de propiedades de contexto disponibles en cada contexto comparado es una cuestión importante. Si se omiten propiedades relevantes en la especificación de cualquiera de los contextos involucrados, el resultado final para el grado de similitud/aplicabilidad será subvaluado –es decir, un valor menor al que se obtendría si se hubieran incluido todas las propiedades relevantes. Esto puede llevar a resultados no confiables y consecuentemente a interpretaciones y decisiones erróneas. Por eso se debe prestar especial atención en la selección de las propiedades de contexto relevantes apropiadas de ambos contextos, reales y de aplicación.

La especificación de la métrica *CApp*, siguiendo las definiciones del marco C-INCAMI, se presenta en la Tabla 5.2.

Tabla 5.2: Especificación de la métrica “Aplicabilidad de Entidad Contextual” (*CApp*) usando las definiciones provistas por el marco C-INCAMI.

INDIRECT METRIC			
Name	Aplicabilidad Contextual (<i>CApp</i>)	Objective	Determinar la aplicabilidad de una entidad contextual a un determinado contexto real.
References	—	Accuracy	100 %
Author	Hernán Molina	Version	1.0
Value interpretation	Grado o proporción en que una entidad es contextualmente aplicable a un determinado contexto real.		
Related Metric	Similitud de Propiedad de Contexto (<i>CPSim</i>)		
SCALE (NUMERICAL SCALE)			
Value type	REAL	Scale type	RATIO
Representation	CONTINUOUS	UNIT	
Range Min Value	0.0	Name	Aplicabilidad a Contexto
Range Max Value	1.0	Description	Grado o nivel de aplicabilidad contextual de una entidad a un contexto real.
Step Value	—	Acronym	ac
FUNCTION			
Name	Fórmula <i>CApp</i>		
Specification	$CApp(TC(e), AC(p)) = \sum_{i=1}^{ CP(e) } CWcp_k(e) \times CPSim(cp_i, TC(e), AC(p))$		
CALCULATION METHOD			
Name	Cálculo de <i>CApp</i>		
Specification	Calcular la Similitud de Propiedad de Contexto (<i>CPSim</i>) para cada propiedad del contexto de aplicación de la entidad contextual respecto de la propiedad equivalente del contexto real y luego calcular la suma de las similitudes ponderadas por el peso de la propiedad respectiva en el contexto de referencia.		
References	—		

5.2.1.3. Similitud de Propiedades de Contexto (*CPSim*)

Como se indicó al principio de la Subsección 5.2.1, la similitud entre los valores especificados para una propiedad de contexto en diferentes descripciones de contexto es determinada o calculada utilizando el *Criterio de Similitud (SimilarityCriterion)* asociado a la misma en el contexto usado como base o referencia (como se indicó en cada métrica descrita arriba). El valor producido por la métrica *CPSim* representa el nivel o grado en que dos propiedades equivalentes (provenientes de una misma definición), incluidas en dos descripciones de contexto diferentes, son similares. El nivel de similitud se calcula en base a la proporción de valores asignados a la propiedad en el primer contexto que son iguales a los valores especificados para la misma propiedad en la segunda descripción de contexto.

Las ecuaciones correspondientes a los criterios de similitud (y sus variantes), que se presentan a continuación, corresponden al cálculo de la métrica *CPSim* incluida en la especificación de las métricas *CSim* (Ecuación 5.1) y *CApp* (Ecuación 5.2). Para cada ecuación, se indicarán las condiciones en las cuales pueden ser aplicadas (tanto de la propiedad como del tipo de contexto a describir –real o de aplicación).

Para el cálculo de similitud de propiedades equivalentes se consideran los siguientes términos:

- De la misma forma que se utilizó $AC(e)$ y $TC(e)$ para representar a un *contexto real* o un *contexto de aplicación* respectivamente, aquí se utilizará $C(e)$ para representar de forma genérica a un *contexto*, pudiendo referirse a cualquiera de los dos primeros.
- como se introdujo en un principio, $CVcp_k(a) = \{cv_1, \dots, cv_m\}$ es el conjunto de los valores especificados para la propiedad de contexto cp_k en el contexto asociado a la entidad a y $CVcp_k(b) = \{cv_1, \dots, cv_n\}$ los valores especificados para la misma propiedad de contexto en la entidad b ;
- se define $CVcp_k(a, b) = CVcp_k(a) \cup CVcp_k(b)$ como todos los valores diferentes especificados para la propiedad de contexto cp_k para ambas entidades a y b en conjunto. Note que los valores de la propiedad en cada entidad provienen de la misma definición de la propiedad, como se explicó antes; y

- $|CVcp_k(a, b)|$ es el número de valores incluidos en $CVcp_k(a, b)$;

Cualquiera sea el criterio de similitud asignado a la propiedad cp_k , la ecuación correspondiente será calculada cuando dicha propiedad esté incluida en la descripción de los dos contextos comparados, de otra forma, la similitud entre los valores de la propiedad cp_k especificados para ambos contextos será 0.

En la Tabla 5.3 se presenta la especificación de la métrica $CPSim$ siguiendo las definiciones del marco C-INCAMI. En este caso se trata de una métrica directa ya que no depende de otras métricas para obtener un valor (como fue definido en el capítulo anterior). Cabe aclarar que en la especificación presentada en dicha tabla no se incluye la especificación del método de medición ya que estos corresponden a los diferentes criterios (y sus variantes) que pueden utilizarse para el cálculo de la similitud de propiedades de contexto, y que serán descriptos a continuación.

Tabla 5.3: Especificación de la métrica “Similitud de Propiedad de Contexto” ($CPSim$) usando la definiciones provistas por el marco C-INCAMI.

DIRECT METRIC			
Name	Similitud de Propiedad de Contexto (CPSim)	Objective	Determinar la similitud entre los valores asignados a dos propiedades equivalentes en diferentes contextos.
References	—	Accuracy	100 %
Author	Hernán Molina	Version	1.0
Value interpretation	Grado o proporción en que los valores asignados a una propiedad en un contexto son similares a los valores asignados a una propiedad equivalente en otro contexto dado.		
SCALE (NUMERICAL SCALE)			
Value type	REAL	Scale type	RATIO
Representation	CONTINUOUS	UNIT	
Range Min Value	0.0	Name	Similitud de Propiedad de Contexto
Range Max Value	1.0	Description	Grado o nivel de similitud entre los valores asignados a dos propiedades en diferentes contextos.
Step Value	—	Acronym	spc

Criterio de Similitud Exacto (ExactSimilarityCriterion):

Este criterio puede aplicarse a propiedades que pueden asumir una cantidad arbitraria de valores (indicado por el atributo *multiplicityOfValues* de *ContextProperty*). La aplicación de este criterio comprueba que cada valor especificado para la propiedad en el primer contexto sea igual a alguno de los valores especificados para la propiedad equivalente en el segundo contexto. El resultado de cada comparación uno a uno es entonces agregado en un resultado final de similitud para la propiedad utilizando un operador lógico (*logicAggregator*) asignado al criterio. El operador lógico puede asumir cuatro valores posibles: AND, XAND, OR o FUZZY. En el caso de los tres primeros operadores, el resultado de similitud será 1 o 0 –verdadero o falso- mientras que para el operador FUZZY el resultado de similitud será un valor de verdad difuso en un rango continuo entre 0 y 1. Si se especifica el operador AND (Ecuación 5.3), las propiedades se consideran similares si se comprueba que todos los valores de la propiedad en el contexto de referencia están presentes en la propiedad equivalente del segundo contexto; si se especifica el operador XAND –eXclusive AND- (Ecuación 5.4), las propiedades se consideran similares si se comprueba que todos los valores de la propiedad en el contexto de referencia están presentes en la propiedad equivalente del segundo contexto y dichos valores son los únicos especificados en ese último contexto; cuando se utiliza el operador OR (Ecuación 5.5) se comprueba que al menos un valor de la propiedad en el contexto de referencia esté presente en la propiedad equivalente del segundo contexto; para el operador FUZZY (Ecuación 5.6) el resultado será la proporción de valores iguales del total de valores diferentes especificadas para la propiedad en sendos contextos.

$$ESC_{AND}(cp_k, C(a), C(b)) = \begin{cases} 1 & \text{si } \forall cv_i \exists cv_j (cv_i \in CVcp_k(a) \wedge cv_j \in CVcp_k(b) \wedge cv_i = cv_j) \\ 0 & \text{en otro caso} \end{cases} \quad (5.3)$$

$$ESC_{XAND}(cp_k, C(a), C(b)) = \begin{cases} 1 & \text{si } \forall cv_i \exists cv_j (cv_i \in CVcp_k(a) \wedge cv_j \in CVcp_k(b) \wedge cv_i = cv_j) \\ & \wedge |CVcp_k(a)| = |CVcp_k(b)| \\ 0 & \text{en otro caso} \end{cases} \quad (5.4)$$

$$ESC_{OR}(cp_k, C(a), C(b)) = \begin{cases} 1 & \text{si } \exists cv_i \exists cv_j (cv_i \in CVcp_k(a) \wedge cv_j \in CVcp_k(b) \wedge cv_i = cv_j) \\ 0 & \text{en otro caso} \end{cases} \quad (5.5)$$

$$ESC_{FUZZY}(cp_k, C(a), C(b)) = \frac{\sum_{i=1}^{|CVcp_k(a)|} VM(cv_i, CVcp_k(b))}{|CVcp_k(a, b)|}, \forall cv_i \in CVcp_k(a) \quad (5.6)$$

donde *VM* (*Value Match*) representa la comparación de un determinado valor cv_i con todos los valores especificados para la propiedad cp_k en el contexto de b , es decir cada valor incluido en $CVcp_k(b)$:

$$VM(cv_i, CVcp_k(b)) = \begin{cases} 1 & \text{si } \exists cv_j (cv_j \in CVcp_k(b) \wedge cv_i = cv_j) \\ 0 & \text{en otro caso} \end{cases}$$

Se debe recordar que, según se indicó en la Subsección 5.1.1, a todas las propiedades de un *contexto real* se asigna por defecto un *criterio de similitud exacta*, con un operador FUZZY. Esta decisión se debe a que el criterio de similitud exacta con el operador FUZZY provee un criterio de comparación de propósito general pensado, y más apropiado, para realizar comparaciones de similitud entre contextos del mismo tipo (dos contextos reales). Mientras que el mismo criterio con los operadores AND, XAND y OR, así como el resto de los criterios son más apropiados para realizar comparaciones más específicas para cada propiedad de contexto, pensadas para diseñar diferentes requerimientos para especificar la aplicabilidad contextual de una determinada entidad reusable –digamos una métrica- a un contexto real. Por lo anterior, en la especificación de los criterios restantes se distingue entre un *contexto real* (AC) y un contexto de aplicación (TC).

Como se mencionó anteriormente, cada variante de cada criterio de similitud puede ser especificada utilizando el marco C-INCAMI como un *método de medición* (*MeasurementMethod*) aplicable a la métrica CPSim (que fue presentada en la Tabla 5.3). Así, el método de medición correspondiente al *criterio de similitud exacto* utilizando el operador *AND* puede ser especificado tal como se muestra en la Tabla 5.4, donde la especificación del método consiste en la expresión que define al criterio correspondiente. La especificación del resto de los criterios y sus variantes es análoga a la presentada en dicha tabla.

Tabla 5.4: Método de medición asociado al *Criterio de Similitud de Exacto* utilizando el operador *AND* (ESC_{AND}).

MEASUREMENT METHOD	
Name	Cálculo de ESC_{AND}
Specification	$ESC_{AND}(cp_k, C(a), C(b)) = \begin{cases} 1 & \text{si } \forall cv_i \exists cv_j (cv_i \in CVcp_k(a) \wedge cv_j \in CVcp_k(b) \wedge cv_i = cv_j) \\ & \wedge CVcp_k(a) = CVcp_k(b) \\ 0 & \text{en otro caso} \end{cases}$ $VM(cv_i, CVcp_k(b)) = \begin{cases} 1 & \text{si } \exists cv_j (cv_j \in CVcp_k(b) \wedge cv_i = cv_j) \\ 0 & \text{en otro caso} \end{cases}$
References	—

Criterio de Similitud de Umbral (*ThresholdSimilarityCriterion*):

Este criterio puede aplicarse a propiedades que pueden asumir un solo valor, tanto en el *contexto de aplicación* como en el *contexto real* –es decir, se debe cumplir que $|CVcp_k(e)| =$

1, $|CVcp_k(a)| = 1$. La aplicación de este criterio comprueba que el valor especificado para la propiedad en el *contexto real* sea mayor o menor al valor umbral del criterio (especificado mediante la relación *threshold* en el diagrama de la Figura 5.2), es decir, al valor asignado a la propiedad en el *contexto de aplicación*, según lo indique el parámetro *matchingDirection* del criterio. Este parámetro puede asumir dos valores posibles:

- *UP*, que indica que el valor de la propiedad en el *contexto de aplicación* es considerado un umbral inferior (Ecuación 5.7) por encima del cual el valor comparado se considera similar;
- *DOWN*, para indicar que dicho valor es considerado un umbral superior (Ecuación 5.8) por debajo del cual el valor comparado se considera similar;

$$TSC_{UP}(cp_k, TC(e), AC(a)) = \begin{cases} 1 & cv_a > cv_{th} \\ 0 & \text{en otro caso} \end{cases} \quad (5.7)$$

$$TSC_{DOWN}(cp_k, TC(e), AC(a)) = \begin{cases} 1 & cv_a < cv_{th} \\ 0 & \text{en otro caso} \end{cases} \quad (5.8)$$

donde, para ambos casos, $cv_a \in CVcp_k(a)$, $cv_{th} \in CVcp_k(e)$.

Criterio de Similitud de Rango (RangeSimilarityCriterion):

Este criterio puede aplicarse a propiedades que pueden asumir un solo valor en el *contexto real*. La aplicación de este criterio comprueba que el valor especificado para la propiedad en el *contexto real* se encuentre dentro del rango especificado por el criterio –tomados de los valores asignados a la propiedad en el *contexto de aplicación*–, es decir, entre un valor umbral superior y un valor umbral inferior (las relaciones *upperThreshold* y *lowerThreshold* en la Figura 5.2, respectivamente). Como se puede ver en la Ecuación 5.9 se deben especificar exactamente dos valores para la propiedad en el contexto de aplicación, de forma que puedan asignarse al criterio ambos umbrales; es decir se debe cumplir: que $|CVcp_k(e)| = 2$, $|CVcp_k(a)| = 1$.

$$RSC(cp_k, TC(a), AC(a)) = \begin{cases} 1 & cv_{lo} < cv_a < cv_{up} \\ 0 & \text{en otro caso} \end{cases} \quad (5.9)$$

donde $cv_a \in CVcp_k(a)$, $cv_{lo}, cv_{up} \in CVcp_k(e)$.

5.2.1.4. Métricas para Comparación de Contextos

Como se indicó al principio de la subsección 5.2.1 las ecuaciones descriptas anteriormente –*Similitud Contextual (CSim)*, *Aplicabilidad contextual (CApp)* y *Similitud de Propiedad de Contexto (CPSim)*– se materializan en el marco C-INCAMI como especificaciones de métricas asociadas apropiadamente a los elementos de información de un proyecto de M&E para implementar los cálculos que permitan obtener los valores que describan, tanto el grado de similitud de cada propiedad de contexto, como del contexto completo en la comparación con una segunda descripción de contexto –ya sea entre dos contextos reales o un contexto de aplicación y uno real. A continuación se describe la forma en que se utiliza cada una de las métricas vistas en el cálculo de comparaciones de similitud entre contextos en proyectos de M&E especificados con C-INCAMI. Para ilustrar mejor el uso de estas métricas, en la Figura 5.8 se presenta un diagrama de objetos UML que representa la información especificada para dos proyectos de M&E hipotéticos, un proyecto actual y otro pasado –éste último de forma resumida, incluyendo sólo los objetos necesarios para ilustrar la situación. La situación representada en el diagrama

es la de una comparación entre dos contextos reales, tomando el contexto del proyecto actual como referencia, comparado contra el contexto del proyecto pasado. En el diagrama se resaltan los objetos (en gris) y las relaciones (en negro) involucrados en la especificación de tales métricas.

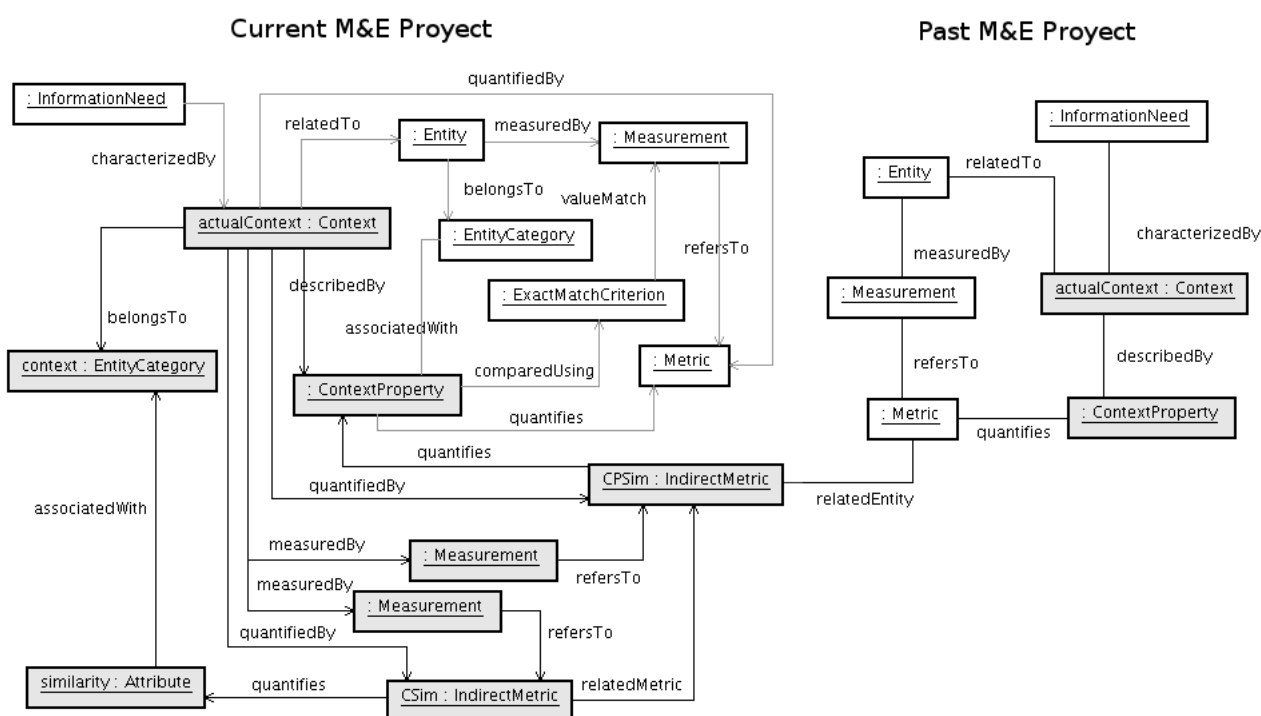


Figura 5.8: Diagrama de objetos (UML) ilustrando el uso de las métricas utilizadas en la comparación de contextos reales entre dos proyectos hipotéticos.

Para cada propiedad incluida en el contexto de referencia, se especifica una métrica *CPSim* como una métrica indirecta asociada (relación *quantifies*) a la propiedad misma, para la cual se calculará el valor de similitud. Cada métrica *CPSim* es agregada a las métricas que cuantifican el contexto de referencia (relación *quantifiedBy*). La especificación de cada métrica *CPSim* se construye a partir del criterio de similitud asociado a la propiedad de contexto cuantificada. Ésta especificación incluye los valores o parámetros asignados al criterio, que corresponden a los valores medidos para la propiedad en el contexto correspondiente. Esta métrica indirecta utiliza como entrada, para el cálculo de similitud (ver relación *relatedMetric*), la métrica que cuantifica a la propiedad de contexto equivalente en el contexto comparado. La métrica *CPSim* puede ser utilizada para calcular la similitud de los valores especificados para una propiedad, tanto entre dos contextos reales, para determinar la similitud contextual de dos proyectos –mediante la métrica *CSim*–, como entre un contexto de aplicación y un contexto real, para calcular la aplicabilidad de una entidad contextual a un determinado proyecto –utilizando la métrica *CApp*. En el primer caso, el contexto de referencia es el *contexto real* del proyecto y cada métrica *CPSim* utilizará como entrada al cálculo la métrica que cuantifica la propiedad equivalente en el *contexto real* del proyecto comparado; en el segundo caso, el contexto de referencia es el *contexto de aplicación* de una entidad contextual que pretende ser utilizada en el proyecto actual (como por ejemplo, una métrica o un modelo elemental de indicador) y cada métrica *CPSim* utilizará como entrada para el cálculo (*relatedMetric*) la métrica que cuantifica la propiedad equivalente en el *contexto real* del proyecto destino. El resultado del cálculo de similitud de esta métrica es registrado por una medición asociada al contexto de referencia (relación *measuredBy*).

Por otro lado, una métrica *CSim* es especificada (utilizando la Ecuación 5.1) como una métrica indirecta asociada al *contexto real* de referencia (relación *quantifiedBy*) en una comparación de similitud contextual. De forma análoga, una métrica *CApp* es especificada (utilizando la Ecuación 5.2) como

una métrica indirecta asociada al *contexto de aplicación* de referencia (relación *quantifiedBy*) en una comparación de aplicabilidad contextual. A diferencia de la métrica *CPSim*, asociada a una propiedad del contexto, la métrica *CSim/CApp* es asignada (relación *quantifies*) a un Atributo "*similitud*" asociado a (relación *associatedWith*) la *Categoría de Entidad "contexto"* a la cual pertenece el objeto *contexto* mismo –ambos definidos con el propósito de mantener la consistencia con el módulo *requirements* sobre el cual se construye el modelo de contexto. Tanto la métrica *CSim* como *CApp* utilizan como entrada para el cálculo de similitud (relación *relatedMetric*) cada una de las métricas utilizadas para calcular la similitud de las propiedades (*CPSim*) incluidas en el contexto de referencia. El resultado del cálculo de similitud de contexto es registrado (en ambos casos, *CSim* y *CApp*) como una medición asignada al contexto de referencia (relación *measuredBy*).

Prueba de Concepto



En este capítulo se desarrolla una prueba de concepto con el objetivo de mostrar la factibilidad del uso del marco C-INCAMI en la especificación de los datos y metadatos involucrados en la medición y evaluación de calidad de entidades de software a partir de las definiciones presentadas en los capítulos anteriores. La prueba de concepto se basa en un caso de estudio [Olsina *et al.*, 2008a] en el que se utilizó el marco INCAMI para especificar la medición y evaluación de un componente web, particularmente el carro de compras de la aplicación web *Cuspide.com*¹. La prueba de conceptos desarrollada a continuación incluye la especificación de al menos una instancia de datos y metadatos para todos los conceptos definidos en el marco (presentados a lo largo del Capítulo 4). En este sentido, se cubren cada una de las etapas que involucra el marco C-INCAMI, considerando tanto la definición del proyecto de M&E, el diseño (definiciones de requerimientos, métricas e indicadores) y la implementación de cada una de las actividades (resultados de medición y evaluación), incluyendo también aquellos datos y metadatos del contexto utilizados en la recomendación sensible al contexto durante la selección de los diferentes elementos de diseño (tales como modelos de concepto, métricas y modelos de indicadores) y los cálculos realizados en cada caso. Se destacan, finalmente, los aspectos más relevantes de la evaluación de los resultados con fines de interpretación, comparación y mejora.

6.1. Creando un Proyecto de Medición y Evaluación

Se comienza por definir un Proyecto de Medición y Evaluación (*MEProject*) en el que se indica el nombre del proyecto, su director (e información de contacto) y fechas de inicio y fin (estimada). En la Tabla 6.1 se representa a dicho proyecto, utilizando los metadatos descriptos en la Figura 4.2 (Sección 4.2).

Tabla 6.1: Datos del Proyecto de M&E.

<i>Name</i>	<i>Description</i>	<i>Director</i>
Evaluation of Cuspide.com	External quality evaluation of Cuspide.com	Hernán Molina
<i>Contact information</i>	<i>Start date</i>	<i>End date</i>
hmolina@ing.unlpam.edu.ar	01/10/2011	15/10/2011

¹Empresa argentina dedicada a la venta on-line de libros [<http://www.cuspide.com>].

A partir de esta instancia del Proyecto de M&E se definen los objetivos mediante una *necesidad de información* y los subproyectos correspondientes a las actividades de medición (*MeasurementProject*) y evaluación (*EvaluationProject*).

6.2. Especificando los Requerimientos No Funcionales

La primer etapa de diseño consiste, como se adelantó antes, en la definición de la *necesidad de información* para el proyecto actual y la correspondiente especificación de los requerimientos no funcionales mediante un *modelo de concepto*.

6.2.1. Necesidad de Información

La especificación de la *necesidad de información* se realiza utilizando los metadatos presentados en la Figura 4.3 (Sección 4.3) estableciendo que (como se representa en la Tabla 6.2) se pretende *mejorar la calidad externa* de una *aplicación web* desde el punto de vista de un *usuario cliente*. Particularmente se evaluará la aplicación web *Cuspide.com*.

Tabla 6.2: La especificación de la *Necesidad de Información* para la evaluación de Calidad externa de la aplicación web Cuspide.com

<i>Purpose</i>	<i>Focus</i>	<i>User Viewpoint</i>	<i>Entity Category</i>	<i>Entity</i>
Improve	External Quality	Customer user	Web Application	Cuspide.com

6.2.1.1. Descripción del Contexto (Real)

También como parte de la especificación de la *necesidad de información* se identifican y cuantifican las propiedades que describen el *contexto real* que la caracteriza. Algunas de las propiedades de contexto que se utilizarán ya se encuentran definidas como tales en el catálogo de contexto, mientras otras son importadas del catálogo de dominio de la organización e incluidas como propiedades de contexto en el catálogo de contexto. Se debe recordar, como se vio en la Sección 3.2.1, que las propiedades pueden ser tomadas del catálogo de dominio o definidas exclusivamente para el catálogo de contexto. Como también se comentó anteriormente en la Sección 5.1.1, la decisión de agregar una propiedad al contexto real es únicamente elección del evaluador a cargo del proyecto de M&E, utilizando la información especificada mediante el marco C-INCAMI a modo de recomendación (ya sean las entidades involucradas en el proyecto o las propiedades de contexto relacionadas entre sí).

Las propiedades de contexto incluidas en el contexto real se presentan en la Tabla 6.3. Para cada propiedad se especifica (siguiendo el modelo de la Figura 5.2 presentado en la Sección 5.1) su *nombre*, *definición*, *relevancia*, *URI de la propiedad* en el catálogo de contexto y la *multiplicidad de valores* que puede asumir cada una. También se especifica la *categoría de entidad* a la cual está asociada la propiedad, la *escala* de la métrica que la cuantifica –todas ellas categóricas en este caso, enumerando sólo los valores permitidos (*CategoricalValue*)–, las *propiedades de contexto relacionadas* y el *valor* asignado (por medio de mediciones, cuyos metadatos no se incluyen). Las métricas completas para algunas de las propiedades se mostrarán luego. No se incluyen en la tabla el *objetivo* de cada propiedad (definido en la clase *Attribute*), ya que en este caso es igual para todas las propiedades del contexto real –“*caracterizar el contexto real relevante de la entidad sometida a M&E*”– ni el *peso*, que se asigna de forma equitativa o balanceada entre todas las propiedades ($0,8\overline{333}$ para cada una, sumando en total 1,0).

Como se indicó en la Subsección 5.1.1, se identifican previamente las entidades relevantes involucradas en el contexto real. En este caso se encuentran dos entidades relevantes: una de ellas es el mismo *Proyecto de M&E* y la otra es la aplicación web siendo evaluada (*Cuspide.com*).

De la entidad *Proyecto de M&E* se consideran tres propiedades como relevantes para el contexto. Todas ellas ya se encontraban definidas en el catálogo de contexto. Una de ellas describe la *categoría de entidad* especificada por la necesidad de información. Esta redundancia es necesaria ya que contar con la categoría de entidad como propiedad de contexto puede ayudar en las recomendaciones de diseño sensible al contexto tanto de los requerimientos como de la medición. La siguiente propiedad, "*Evaluation target*", describe el ámbito en el cual se interpretarán y utilizarán los resultados de la evaluación, en este caso, siendo utilizados sólo en el ámbito interno de la organización. La tercer propiedad de contexto del *proyecto de M&E* describe el *soporte tecnológico* disponible en el mismo para la automatización de diversos métodos que pudieran ser utilizados durante la implementación de la medición y la evaluación de la entidad *Cuspide.com*. En este caso, el proyecto cuenta con herramientas para automatizar el método *LSP* [Dujmovic, 2007], la métrica de legibilidad *Lexile*² y un conjunto de herramientas análisis para sitios web .

El resto de las propiedades encontradas como relevantes para el contexto pertenecen a la categoría de entidad *Web Application*:

- La propiedad "*Application domain*", aunque se recupera del catálogo de contexto, se encuentra definida también en el catálogo de dominio de la organización (del cual se obtuvo originalmente). Dicha organización se especializa en el desarrollo de aplicaciones de software en cuatro dominios (e-commerce, e-government, e-banking, advertisement –complementados con el servicio adicional de correo electrónico web propio), tal como se refleja en la escala de la métrica que cuantifica la propiedad de contexto en cuestión.
- La propiedad "*Target market scope*" se encuentra definida únicamente en el catálogo de contexto ya que no es una propiedad que se haya definido previamente en el dominio de aplicación. Esta propiedad permite describir si el mercado al que apunta el sitio es de ámbito nacional, regional, o internacional (tal como lo refleja la escala de la métrica que cuantifica la propiedad). Esta propiedad tiene asociadas otras tres propiedades de contexto –"*Website owner's country*", "*Target geographic region*" y "*Supported natural language*"- que son incluidas en la descripción del contexto real.
- La propiedad "*Website owner's country*", en relación a la propiedad anterior, permite indicar el país en el que reside el propietario de la aplicación web y, junto con la propiedad "*Target geographic region*" –que indica la/s región/es o continente/s donde reside la audiencia de la aplicación- establecen un marco de referencia contextual respecto de la situación geográfica del sitio web, que puede ser utilizada para recomendar la selección de determinados atributos para especificar los requerimientos de la evaluación y/o métricas para cuantificar a dichos atributos.
- También relacionadas a las propiedades anteriores, la propiedad "*Supported natural language*" permite describir los lenguajes naturales en los que puede ser accedida la aplicación web, también útil en la selección de métricas que se apliquen sobre el texto del sitio (contenido) y que pudieran depender del lenguaje utilizado por el mismo.
- Al igual que "*Application domain*", la propiedad "*Software release life-cycle stage*" se encuentra definida en el catálogo de dominio de la organización. Ésta es utilizada para describir la etapa del ciclo de vida de software en la que se encuentra la entidad siendo evaluada, considerando que se trata de un producto (una aplicación web en este caso). La entidad *Cuspide.com* se encuentra en la etapa "*Live Release*" ya que se trata de un producto final actualmente en uso.
- La propiedad "*Content provider role*" es definida exclusivamente para el catálogo de contexto e incluida en el contexto real para describir el rol de quien es el proveedor del contenido principal

²<http://www.lexile.com/about-lexile/lexile-overview/>

de la aplicación web. Esta propiedad se relaciona a la propiedad “*Main content type*” (también una propiedad definida únicamente en el catálogo de contexto) que permite identificar el tipo de contenido principal del sitio web. En el caso de *Cuspide.com* se trata de “*texto e imágenes*” (tanto para la descripción del contenido de los libros como para ilustrar la tapa o vistas previas de los mismos).

- La propiedad “*Traded product type*” es incluida debido al dominio de la aplicación web (según se especifica en la propiedad “*Application domain*”) pudiendo ser utilizada para recomendar opciones de diseño tanto para la medición como para la evaluación de aquellos atributos incluidos (directa o indirectamente) bajo el concepto “*Information Quality*”. En el caso de *Cuspide.com* sólo se indica el valor “*books*” como tipo de producto comercializado.

Tabla 6.3: Las propiedades del contexto real que caracteriza la necesidad de información del proyecto.

Name	Entity category	Entity Category	M&E Project
Definition	Category to which the evaluated entity belongs.	Related Context Property	Evaluation target
Relevance	The entity category may determine other relevant properties to be used in the context description. It may also help in selecting information elements during M&E design .	Categorical Scale (Categorical Values)	[Web application Desktop application Embedded component Web page Test case Application server]
		Multiplicity Of Values	0
Prop. URI	http://myorg.com/context/EntityCategory	Measures (Value)	Web Application
Name	Evaluation target	Entity Category	M&E Project
Definition	The scope or range within which evaluation results are interpreted for decision making.	Related Context Property	–
Relevance	It may help in selecting information elements during evaluation design.	Categorical Scale (Categorical Values)	[internal evaluation external certification]
		Multiplicity Of Values	1
Prop. URI	http://myorg.com/context/EvaluationTarget	Measures (Value)	internal evaluation
Name	Automated support	Entity Category	M&E Project
Definition	Support for the automation of different methods and procedures to be used in the implementation of M&E activities.	Related Context Property	–
Relevance	It may help in selecting information elements during M&E design that rely on automated methods.	Categorical Scale (Categorical Values)	[Additive Model LSP Method Lexile Analyzer® Web crawler]
		Multiplicity Of Values	0
Prop. URI	http://myorg.com/context/AutomatedSupport	Measures (Value)	LSP method, Lexile Analyzer®, Web crawler
Name	Application domain	Entity Category	Web Application
Definition	Domain to which the web application belongs.	Related Context Property	–
Relevance	It may help in selecting information elements during requirements and measurement design (such as concept models, attributes and metrics).	Categorical Scale (Categorical Values)	[e-mail e-commerce e-government e-banking advertisement]
		Multiplicity Of Values	1
Prop. URI	http://myorg.com/context/ApplicationDomain	Measures (Value)	e-commerce
Name	Target market scope	Entity Category	Web Application
Definition	The geographic scope or range of the market to which the web application is aimed.	Related Context Property	Website owner’s country, Target geographic region, Supported natural language.
Relevance	It may affect outcomes interpretation regarding <i>Information Quality</i> , particularly <i>Coverage (Suitability)</i> or in the selection of attributes related to those concepts	Categorical Scale (Categorical Values)	[national regional international]
		Multiplicity Of Values	1
Prop. URI	http://myorg.com/context/TargetMarketScope	Measures (Value)	international
Name	Website owner’s country	Entity Category	Web Application
Definition	The country in which the owner of the web application resides.	Related Context Property	Target market scope, Target geographic region, Supported natural language.
Relevance	It may affect outcomes interpretation regarding <i>Information Quality</i> , particularly <i>Coverage (Suitability)</i> .	Categorical Scale (Categorical Values)	An enumeration of country names.
		Multiplicity Of Values	1
Prop. URI	http://myorg.com/context/WebsiteOwnerCountry	Measures (Value)	Argentina

Name	Target geographic region	Entity Category	Web Application
Definition	Geographic region in which the targeted audience resides.	Related Context Property	Target market scope, Website owner's country, Supported natural language
Relevance	It may affect outcomes interpretation regarding <i>Information Quality</i> , particularly <i>Coverage (Suitability)</i> .	Categorical Scale (Categorical Values)	[North America South America Europe Asia Africa Australia Antarctica]
Prop. URI	http://myorg.com/context/TargetGeographicRegion	Multiplicity Of Values	0
		Measures (Value)	South America, North America, Europe, Asia, Africa, Australia

Name	Supported natural language	Entity Category	Web Application
Definition	Natural languages in which the content of the Web site is written.	Related Context Property	Target market scope, Website owner's country, Target geographic region, Main content type
Relevance	It may help in measurement design since the Web site's text is part of the content whose quality will be measured and evaluated	Categorical Scale (Categorical Values)	An enumeration of natural languages' names.
Prop. URI	http://myorg.com/context/SupportedNaturalLanguage	Multiplicity Of Values	0
		Measures (Value)	Spanish

Name	Software release life-cycle stage	Entity Category	Web Application
Definition	Stage that describes the stability of a released piece of software as the development process proceeds.	Related Context Property	–
Relevance	May affect outcomes interpretation since different quality levels may be observed at different stages.	Categorical Scale (Categorical Values)	[Pre-alpha Alpha Beta Release Candidate Release to Marketing General Availability Live Release] ³
Prop. URI	http://myorg.com/context/SoftwareReleaseLifeCycleStage	Multiplicity Of Values	1
		Measures (Value)	Live Release

Name	Content provider role	Entity Category	Web Application
Definition	Role of the user that provides the main content of the Web application.	Related Context Property	–
Relevance	May affect outcomes interpretation since different quality levels may be observed on content provided by different types of providers.	Categorical Scale (Categorical Values)	[owner registered user].
Prop. URI	http://myorg.com/context/ContentProviderRole	Multiplicity Of Values	1
		Measures (Value)	owner

Name	Main content type	Entity Category	Web Application
Definition	The type of the main content of the web site.	Related Context Property	Supported Natural Language
Relevance	It may help in requirements, measurement and evaluation design as well as in the interpretation of M&E results.	Categorical Scale (Categorical Values)	[Text Images Plugin]
Prop. URI	http://myorg.com/context/MainContentType	Multiplicity Of Values	0
		Measures (Value)	Text, Images

Name	Traded products type	Entity Category	Web Application
Definition	Type of products the e-commerce web site trades.	Related Context Property	–
Relevance	It may help in measurement design since products properties are part of the content whose quality will be measured and evaluated.	Categorical Scale (Categorical Values)	[books music movies electronics games clothes home office sports]
Prop. URI	http://myorg.com/context/TradedProductType	Multiplicity Of Values	0
		Measures (Value)	books

A cada una de las propiedades incluidas en el contexto real se le asigna además el criterio de similitud correspondiente, según se indicó en la Subsección 5.1.1, es decir, un *criterio de similitud exacto* con un operador *FUZZY*. En la Tabla 6.4 se muestra el criterio de similitud para la propiedad “*Automated Support*”.

A continuación se describen las métricas que cuantifican algunas de las propiedades del contexto real. Estas métricas se especifican de la misma forma que las utilizadas en la medición de requerimientos no funcionales, tal como se describió en la Sección 4.4. No obstante cabe destacar algunas particularidades al respecto. Si una propiedad de contexto fue obtenida originalmente del catálogo de dominio de la organización, el método de medición de la métrica que la cuantifica puede especificar que los valores se obtienen mediante el procedimiento asociado al mecanismo de integración al dominio de aplicación

³Tomado de http://en.wikipedia.org/wiki/Software_release_life_cycle.

Tabla 6.4: Criterio de similitud para la propiedad de contexto “Automated Support”.

Exact Similarity Criterion	
Logic aggregator	FUZZY
Value Match	LSP method
Value Match	Lexile metric
Value Match	Web site analysis

del marco C-INCAMI, basado en RDF (explicado en la Sección 4.6). En este caso, la *especificación del método de medición (MeasurementMethod)* es la siguiente:

“Efectuar una consulta RQL al catálogo de dominio utilizando como parámetro una *tripla RDF* en la que el *sujeto* sea el valor *individualURI* de la entidad (*Entity*) a la que pertenece el atributo (*Attribute*) a cuantificar y el *predicado* sea el valor *propertyURI* de dicho atributo. El valor medido corresponde al nombre del valor categórico (*CategoricalValue*) cuyo *individualURI* corresponda al URI instanciado en el *objeto* de la tripla como resultado de la consulta.”

Por otro lado, si la propiedad hubiera sido especificada por primera vez en el catálogo de contexto, y no obtenida del catálogo de dominio de la organización, el método debe especificar de forma objetiva (independientemente de la implementación del catálogo) cómo se obtienen los valores.

La métrica que cuantifica la propiedad de contexto “*Application domain*” (esta última especificada originalmente en el catálogo de dominio) es un ejemplo del caso mencionado arriba en que el método de medición especifica la consulta RQL al catálogo RDF. Esta métrica es representada en la Tabla 6.5 (obviando el método de medición por lo indicado anteriormente). En la implementación del método de esta métrica se instancia el *sujeto* de la consulta SQL indicada arriba con “*org:Cuspide.com*” y el *predicado* con “*org:ApplicationDomain*”, obteniendo como resultado el *objeto* instanciado con el URI “*org:e_commerce*” que corresponde al valor categórico “*e-commerce*”. Por otro lado, la referencia especificada para esta métrica indica, de forma abreviada, a la propiedad RDF correspondiente definida en el catálogo de dominio, cuyo URI completo es <http://myorg.com/business-domain/ApplicationDomain> ya que la métrica es generada automáticamente para cuantificar la propiedad correspondiente. Además, ya que la métrica fue generada automáticamente, el autor de la métrica es la herramienta *C-INCAMI Project Manager* que implementa el marco homónimo (la cual se describe en el Capítulo 7).

Tabla 6.5: Métrica que cuantifica la propiedad de contexto “*Application domain*”.

DIRECT METRIC			
Name	Application Domain Name	Objective	To characterize the actual context in which an M&E Project is executed.
References	org:ApplicationDomain	Accuracy	100 %
Author	C-INCAMI Project Manager	Version	1.0
Value interpretation	A categorical symbol which distinguishes the business area covered by a software application.		
SCALE (CATEGORICAL SCALE)			
Value type	SYMBOL	Scale type	NOMINAL
Allowed values	[e-mail e-commerce e-government e-banking advertisement]		

Para la propiedad de contexto “*Entity category*”, la métrica que la cuantifica (representada en la Tabla 6.6) especifica una escala categórica que incluye como valores permitidos las categorías de aquellas entidades que la organización evalúa regularmente durante su actividad. La referencia indicada por esta métrica corresponde al concepto *Entity Category* especificado en los metadatos del marco C-INCAMI

almacenados en el catálogo RDF de M&E, cuyo URI completo es <http://myorg.com/c-incami/EntityCategory>. En este caso, el método de medición de esta métrica indica que el valor para la propiedad se establece consultando la categoría de entidad especificada como parte de la necesidad de información.

Tabla 6.6: Métrica que cuantifica la propiedad de contexto “*Entity category*”.

DIRECT METRIC			
Name	Entity Category Name	Objetive	To characterize the actual context in which an M&E Project is executed.
References	cincami:EntityCategory	Accuracy	100 %
Author	Hernán Molina	Version	1.0
Value interpretation	A categorical symbol which distinguishes the category of the entity being evaluated.		
SCALE (CATEGORICAL SCALE)			
Value type	SYMBOL	Scale type	NOMINAL
Allowed values	[Web application Desktop application Embeeded component Web page Test case Application server]		
MEASUREMENT METHOD			
Name	Determine Entity Category		
Specification	The value for the quantified attribute is determined by observing the name of entity category specified in the Information Need specification of the corresponding M&E Project.		
References	see the Information Need specified for the M&E Project		

Para la propiedad de contexto “*Target market scope*” se especifica la métrica “*Market Scope*” (presentada en la Tabla 6.7). Esta métrica indica como referencia la definición de la propiedad que cuantifica, almacenada en el catálogo de contexto y cuyo URI completo es <http://myorg.com/context/TargetMarketScope>. El método de medición asociado indica consultar la información de envío para determinar si el mercado al que apunta el sitio es nacional, regional, o internacional (de acuerdo a la escala categórica asociada a la métrica). Éste método es similar al asociado a la métrica que cuantifica la propiedad “*Target geographic region*”.

Tabla 6.7: Métrica que cuantifica la propiedad de contexto “*Target market scope*”.

DIRECT METRIC			
Name	Market scope	Objetive	To characterize the actual context in which an M&E Project is executed.
References	org-ctx:TargetMarketScope	Accuracy	100 %
Author	Hernán Molina	Version	1.0
Value interpretation	A categorical symbol which distinguishes the range or scope of the target market of a web application relative to the home country.		
SCALE (CATEGORICAL SCALE)			
Value type	SYMBOL	Scale type	NOMINAL
Allowed values	[national regional international]		
MEASUREMENT METHOD			
Name	Determine Market scope		
Specification	The value for the quantified attribute is determined by observing the countries to which shippings can be delivered: “national” if shippings are only delivered to the home country; “regional” if shippings can be delivered to countries within the continent; “international” if shippings can be delivered worldwide.		
References	None		

Para la propiedad de contexto “*Supported natural language*” se especifica la métrica “*Natural Languages Names*” (presentada en la Tabla 6.8) cuya escala categórica incluye los nombres de los lenguajes naturales relevantes al dominio de la organización. La referencia indicada por la métrica corresponde a la definición de la propiedad cuantificada, almacenada en el catálogo de contexto y cuyo URI completo es <http://myorg.com/context/SupportedNaturalLanguage>. El método de medición asociado indica que los valores a asignar a la propiedad corresponden a los nombres de los lenguajes naturales que soporta la aplicación de forma efectiva.

Tabla 6.8: Métrica que cuantifica la propiedad de contexto “*Supported natural language*”.

DIRECT METRIC			
Name	Natural Languages Names	Objective	To characterize the actual context in which an M&E Project is executed.
References	org-ctx:SupportedNaturalLanguage	Accuracy	100 %
Author	Hernán Molina	Version	1.0
Value interpretation	Categorical symbols identifying the natural languages supported by a web application.		
SCALE (CATEGORICAL SCALE)			
Value type	SYMBOL	Scale type	NOMINAL
Allowed values	[Spanish English French Portuguese Italian ...]		
MEASUREMENT METHOD			
Name	Determine Natural Languages Names		
Specification	The value for the quantified attribute is determined by observing the natural languages in which the web application can be effectively accessed by activating the corresponding option. If no language option is present, the supported language is that in which the web application is accessed by default. In any case, the values to be assigned are the names of each supported natural language.		
References	None		

6.2.2. Modelo de Concepto (Requerimientos No Funcionales)

A continuación se deben especificar los requerimientos no funcionales propiamente dichos para la entidad a evaluar desde el punto de vista especificado. Para esto se especifica un modelo de concepto (*ConceptModel*) en la forma de un árbol de requerimientos en el que el concepto foco de la evaluación –en este caso, *calidad externa*– es el requerimiento de alto nivel que debe descomponerse en conceptos de más bajo nivel y, finalmente, en atributos medibles sobre la entidad.

Como se indicó en la Sección 4.3, el modelo de concepto puede definirse desde cero, seleccionarse de un conjunto predefinido de modelos o una combinación de los anteriores, es decir, se selecciona un modelo predefinido y se modifica para satisfacer las necesidades particulares de la evaluación. Ésta última opción es la que se aplica en este caso. Y ya que el modelo de concepto es una entidad contextual (como se vio también en la Sección 4.3) es posible utilizar el mecanismo de recomendación sensible al contexto para seleccionar el modelo más apropiado al contexto real.

6.2.2.1. Selección de un modelo de concepto

En el catálogo de M&E se encuentran tres modelos de concepto de *Calidad Externa*, incluyendo sólo las características o conceptos calculables (ver Figura 6.1), cada uno de los cuales tiene asociado la correspondiente descripción del contexto de aplicación (ver Tabla 6.9). El primero de los modelos (Figura 6.1a) incluye los conceptos *Usability*, *Functional Suitability*, *Information Quality*⁴ [Lew et al., 2010], *Security* y *Reliability*, siendo aplicable a contextos en los que la *categoría de entidad* a evaluar sea una “*aplicación web*” para el *dominio de aplicación* de “*e-learning*” (Tabla 6.9a). El segundo modelo (Figura 6.1b) se diferencia del primero sólo en que incluye el concepto *Legal Compliance* (código 3.3) bajo *Information Quality* y el concepto *Authenticity* (código 4.2) bajo el concepto *Security*. Estas diferencias lo hacen más apropiado para ser utilizado en “*aplicaciones web*” cuyo *dominio de aplicación* sea “*e-commerce*” o “*e-banking*” (Tabla 6.9b). Por último, el tercer modelo (Figura 6.1c) incluye sólo los conceptos *Usability*, *Information Quality*, *Reliability* siendo aplicable a contextos donde se evalúen “*sitios web*” genéricos –que no requieren aspectos de funcionalidad específica o de seguridad como en los dos casos anteriores (Tabla 6.9c).

A continuación se procede a realizar los cálculos correspondientes al mecanismo de recomendación sensible al contexto [Molina et al., 2010]. Para ello se calcula la aplicabilidad contextual CA_{App} entre el contexto de aplicación de cada modelo de concepto $TC(e_i)$ (presentados en la Tabla 6.9) y el contexto

⁴Definido en Lew et al., 2010 como “*the degree to which the software/WebApp provides accurate, suitable, accessible and legally compliant information.*”

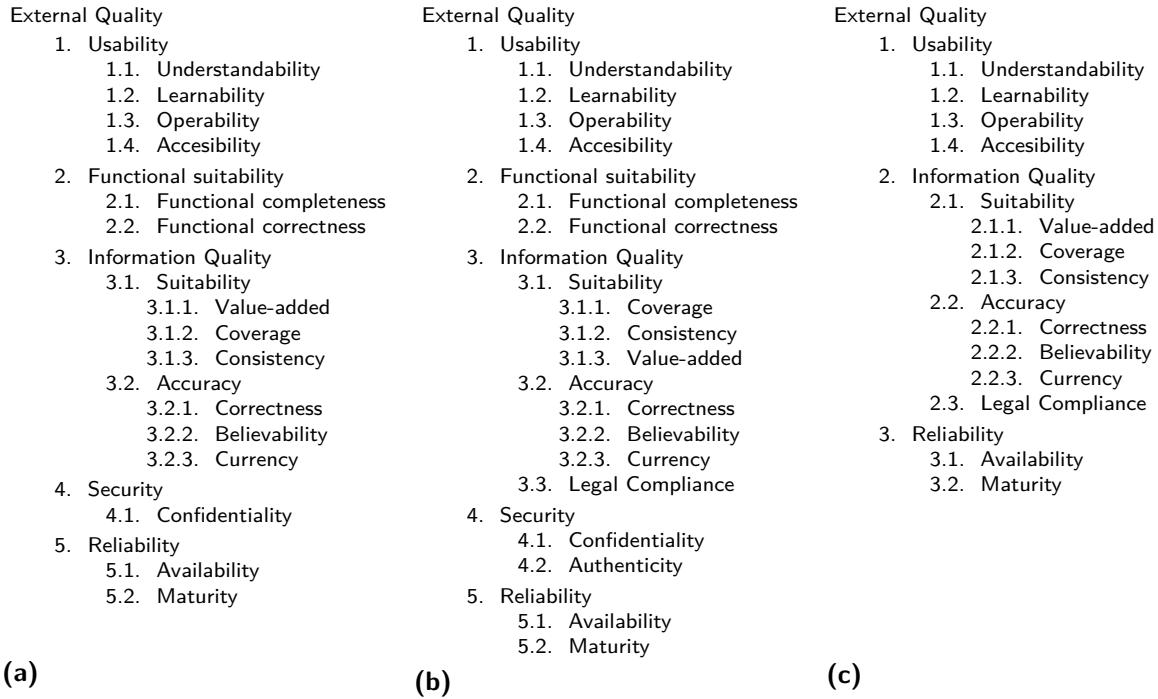


Figura 6.1: Modelos de concepto para Calidad Externa disponibles en el repositorio de la organización.

real del proyecto de M&E $AC(p)$ (de la Tabla 6.3) utilizando la ecuación presentada en la sección 5.2.1.2 y relacionadas:

$$CApp(TC(e), AC(p)) = \sum_{i=1}^{|CP(e)|} CWcp_k(e) \times CPSim(cp_i, TC(e), AC(p))$$

Comenzando por el modelo de concepto de la Figura 6.1a (e_1), cuyo contexto de aplicación $TC(e_1)$ se presenta en la Tabla 6.9a, se calcula la métrica de aplicabilidad contextual de la siguiente forma:

$$CApp(TC(e_1), AC(p)) = (CWentity_category(e_1) \times CPSim(entity_category, TC(e_1), AC(p)) + CWapplication_domain(e_1) \times CPSim(application_domain, TC(e_1), AC(p)))$$

Recordando que $CWentity_category(e_1)$ y $CWapplication_domain(e_1)$ son los pesos especificados para las propiedades “Entity Category” y “Application Domain” respectivamente en el contexto de la entidad e_1 . Se determina primero la similitud de propiedad de contexto $CPSim$ para cada propiedad cp_k en el contexto de aplicación de la entidad contextual $TC(e_1)$ con la propiedad equivalente del contexto real del proyecto actual $AC(p)$. Para la propiedad “Entity Category” se calcula la ecuación correspondiente al criterio de similitud asignado a la misma en el contexto de aplicación (ver Tabla 6.9a). En este caso, ESC_{OR} , para el cual se comprueba que alguno de los valores especificados para dicha propiedad en el contexto de aplicación de la entidad contextual $TC(e_1)$ se encuentre entre los valores especificados para la misma propiedad en el contexto real del proyecto actual $AC(p)$:

$$CPSim(entity_category, TC(e_1), AC(p)) = ESC_{OR}(entity_category, TC(e_1), AC(p)) = \begin{cases} 1 & \text{si } \exists cv_i \exists cv_j (cv_i \in CEntity_category(e_1) \wedge cv_j \in CEntity_category(p) \wedge cv_i = cv_j) \\ 0 & \text{en otro caso} \end{cases}$$

Como puede observarse en las Tablas 6.3 y 6.9a, el valor que cumple con la condición es “Web Application”, ya que se encuentra especificado para la propiedad “Entity Category” tanto en el contexto real del proyecto de M&E como en el contexto de aplicación del modelo de concepto de la Figura 6.1a, por lo tanto

$$CPSim(entity_category, TC(e_1), AC(p)) = ESC_{OR}(entity_category, TC(e_1), AC(p)) = 1$$

Tabla 6.9: Contextos de aplicación de los Modelos de Concepto de la Figura 6.1.

(a) Contexto de aplicación del Modelo de Concepto de la Figura 6.1a

CONTEXT PROPERTY			
Name	Entity Category	Weight	0,5
Relevance	This concept model can only by applied to a particular category of entities.	Multiplicity of Values	0
		Measures (value)	Web Application
Exact Similarity Criterion		Logic Aggregator	OR
		Value Match	Web Application
CONTEXT PROPERTY			
Name	Application Domain	Weight	0,5
Relevance	This concept model can only by applied to a particular application domain.	Multiplicity of Values	0
		Measures (value)	e-learning
Exact Similarity Criterion		Logic Aggregator	OR
		Value Match	e-learning

(b) Contexto de aplicación del Modelo de Concepto de la Figura 6.1b

CONTEXT PROPERTY			
Name	Entity Category	Weight	0,5
Relevance	This concept model can only by applied to a particular category of entities.	Multiplicity of Values	0
		Measures (value)	Web Application
Exact Similarity Criterion		Logic Aggregator	OR
		Value Match	Web Application
CONTEXT PROPERTY			
Name	Application Domain	Weight	0,5
Relevance	This concept model can only by applied to a particular application domain.	Multiplicity of Values	0
		Measures (value)	e-commerce, e-banking
Exact Similarity Criterion		Logic Aggregator	OR
		Value Match	e-commerce
		Value Match	e-banking

(c) Contexto de aplicación del Modelo de Concepto de la Figura 6.1c

CONTEXT PROPERTY			
Name	Entity Category	Weight	1,0
Relevance	This concept model can be applied to a general category of entities.	Multiplicity of Values	0
		Measures (value)	Web Site
Exact Similarity Criterion		Logic Aggregator	OR
		Value Match	Web Site

Luego se determina la similitud de propiedad de contexto $CPSim$ para la propiedad "Application Domain" en el contexto de aplicación del modelo de concepto de la Figura 6.1a (Tabla 6.9a) y el contexto real del proyecto de M&E (Tabla 6.3) usando también el criterio de similitud ESC_{OR} según lo indica la tabla mencionada, de la misma forma que para la propiedad "Entity Category". En este caso, observando ambas tablas, vemos que no existe un valor en común para dicha propiedad en ambos contextos comparados, por lo tanto

$$CPSim(application_domain, TC(e_1), AC(p)) = ESC_{OR}(application_domain, TC(e_1), AC(p)) = 0$$

Luego, la aplicabilidad contextual $CApp$ entre el contexto de aplicación $TC(e_i)$ y el contexto real del proyecto de M&E $AC(p)$ es:

$$CApp(TC(e_1), AC(p)) = (0,5 \times 1 + 0 \times 1) = 0,5$$

es decir el modelo de concepto de la Figura 6.1a es contextualmente aplicable al proyecto de M&E en un 50 %.

Luego se calcula la métrica de aplicabilidad contextual para el modelo de concepto de la Figura 6.1b (e_2), cuyo contexto de aplicación $TC(e_2)$ se presenta en la Tabla 6.9b, de la siguiente forma:

$$CApp(TC(e_2), AC(p)) = (CW_{entity_category}(e_2) \times CPSim(entity_category, TC(e_2), AC(p)) + CW_{application_domain}(e_2) \times CPSim(application_domain, TC(e_2), AC(p)))$$

En el cálculo de la métrica $CPSim$ para la propiedad "Entity Category" (utilizando el criterio de similitud ESC_{OR}) el valor especificado en el contexto de aplicación es igual al especificado en el

contexto real del proyecto de M&E; en el cálculo de $CPSim$ para la propiedad “*Web Application*” uno de los valores especificados en el contexto de aplicación es también igual al especificado en el contexto real; por lo tanto, la métrica $CApp$ para el segundo modelo de concepto se calcula de la siguiente forma:

$$CApp(TC(e_2), AC(p)) = (0,5 \times 1 + 0,5 \times 1) = 1$$

es decir el modelo de concepto de la Figura 6.1b es contextualmente aplicable al proyecto de M&E en un 100 %.

Siguiendo los mismos cálculos para el modelo de concepto de la Figura 6.1c (e_3) se llega a la siguiente ecuación y valor:

$$CApp(TC(e_3), AC(p)) = (0 \times 1) = 0$$

es decir que la aplicabilidad de este modelo de concepto al contexto del proyecto es de 0 %.

ya que el contexto de aplicación de dicho modelo solo incluye una única propiedad de contexto, cuyo valor no se encuentra especificada para la misma propiedad en el contexto real del proyecto de M&E.

Resumiendo los cálculos anteriores, se obtienen los niveles de aplicabilidad contextual al proyecto de M&E para los modelos de concepto de Calidad Externa encontrados en el catálogo de M&E (mostrados en la Tabla 6.10). A partir de estos resultados se determina que el modelo de concepto más aplicable es el modelo de concepto de la Figura 6.1b el cual es seleccionado para describir los requerimientos no funcionales para la necesidad de información del proyecto actual.

Tabla 6.10: Niveles de aplicabilidad contextual al proyecto de M&E para los modelos de concepto de *Calidad Externa* encontrados en el catálogo de M&E.

Modelo de concepto	Aplicabilidad contextual al proyecto de M&E
Figura 6.1a	50 %
Figura 6.1b	100 %
Figura 6.1c	0 %

6.2.2.2. Diseño del Modelo de Concepto (Selección de Atributos)

Hasta aquí se ha seleccionado el modelo de concepto que representará los requerimientos funcionales para la necesidad de información. Sin embargo, como se comentó y vio más arriba, el modelo sólo incluye conceptos calculables por lo que el evaluador debe incluir atributos medibles de las entidades involucradas para poder llevar a cabo la medición y evaluación correspondiente.

Ya que el modelo de concepto fue predefinido para satisfacer un conjunto de requerimientos genéricos, en un contexto también genérico, el evaluador debe analizar primero si todos los conceptos incluidos son útiles y aplicables al caso de la necesidad de información actual. De éste análisis surge que el concepto *Accuracy* (código 3.2 de la Figura 6.1b) –definido como “*The capability of a Web product to deliver information that is correct, credible and current*” [Olsina *et al.*, 2009]– no es aplicable a la necesidad de información especificada ya que en la evaluación que se llevará a cabo no se utilizan técnicas o métodos que permitan realizar comprobaciones de correctitud, credibilidad y actualidad (por ejemplo, la compra efectiva de productos en el sitio web o encuestas a clientes reales). Por lo tanto este concepto (junto a sus subconceptos) es eliminado del modelo. No obstante, esta situación es una oportunidad para incorporar una nueva propiedad al contexto de aplicación del modelo de concepto seleccionado originalmente que refleje el uso de tales métodos, por ejemplo con una propiedad *Measurement Technique* con el valor “*Customer survey*” para reflejar la técnica que permitiría medir atributos

asociados al concepto *Accuracy*. De esta forma, la experiencia obtenida por la modificación del modelo de concepto seleccionado queda registrada en la forma de un contexto de aplicación asociado a la entidad contextual correspondiente, pudiendo ser aprovechada en el futuro.

Se procede entonces a la selección de atributos para cada uno de los conceptos del nivel más bajo del modelo, como se describió en la Sección 4.3. Se debe recordar que, en la Sección 4.3.1 se estableció que un *Atributo* es una entidad contextual, es decir que puede o no ser apropiado para evaluar una determinada entidad dependiendo del contexto relevante a la necesidad de información. Así, la especificación de un atributo en el catálogo de M&E puede estar asociada a una descripción de su contexto de aplicación y, por lo tanto, puede usarse esta información para recomendar los atributos contextualmente aplicables al proyecto actual usando el mecanismo de recomendación sensible al contexto descrito en la Sección 5.2.1.

Asociado al concepto *Usability>Understandability* se encuentra en el catálogo de M&E el atributo *Shopping cart control ease to be recognized*, cuya especificación se presenta en la Tabla 6.11a, que permite determinar si el control para acceder al carro de compras de la aplicación web es fácilmente reconocible. El contexto de aplicación de este atributo es el que se muestra en la Tabla 6.11b donde se especifica que se aplica (de forma genérica) a las aplicaciones web en el dominio de *e-commerce*, las cuales poseen un carro de compras para llevar a cabo la compra de productos, componente sobre el cual se basa la definición del atributo. Muchos de los atributos que se incluyen en el modelo de concepto diseñado especifican un contexto de aplicación similar.

Tabla 6.11: Atributo “Shopping cart control ease to be recognized”.

(a) Especificación del Atributo “Shopping cart control ease to be recognized”.

Name	Shopping Cart Control Ease to be Recognized
Definition	A representation of how easily the control to access the shopping cart in an e-commerce application can be located and recognized as such.
Objective	To determine how easily the control to access the shopping cart in an e-commerce application can be located and recognized as such.
PropertyURI	org:ShoppingCartControlEaseToBeRecognized
Entity Category	Web Application

(b) Contexto de aplicación del Atributo “Shopping cart control ease to be recognized”.

CONTEXT PROPERTY			
Name	Entity Category	Weight	0,5
Relevance	This attribute can be applied to a particular entity category.	Multiplicity of Values Measures (value)	0 Web Application
Exact Similarity Criterion		Logic Aggregator Value Match	OR Web Application
CONTEXT PROPERTY			
Name	Application Domain	Weight	0,5
Relevance	This attribute can be applied to web application domains where a shopping cart is included.	Multiplicity of Values Measures (value)	0 e-commerce
Exact Similarity Criterion		Logic Aggregator Value Match	OR e-commerce

Bajo el concepto *Usability>Learnability* se encuentra el atributo *Readability of text*, cuya especificación se presenta en la Tabla 6.12a, utilizado para determinar la calidad de escritura del texto incluido en la página de ayuda del sitio web. Este atributo es aplicable a aplicaciones, sitios o páginas web cuyo contenido principal incluya texto (independientemente o no de incluir también imágenes), tal como lo refleja la descripción del contexto de aplicación representada en la Tabla 6.12b.

Bajo el concepto *Information Quality>Suitability>Coverage* se encuentra el atributo *Identification of content provider*, cuya especificación se representa en la Tabla 6.13a, utilizado para determinar si el contenido de una aplicación web provisto por usuarios registrados identifica al autor de cada pieza de información publicada. Este atributo es aplicable en contextos donde la categoría de entidad sea una aplicación web, cuyo dominio sea *redes sociales*, *publicación de archivos* o *blogs*, y donde el contenido

del sitio sea provisto por usuarios registrados (como se representa en la Tabla 6.13b).

Tabla 6.12: Atributo “Readability of text”.

(a) Especificación del Atributo “Readability of text”.

Name	Readability of Text
Definition	The quality of written language that makes it easy to read and understand ⁵ .
Objective	To determine the readability of the text included in a web site, application or page.
PropertyURI	org:ReadabilityOfText
Entity Category	Web Site, Web Application, Web Page

(b) Contexto de aplicación del Atributo “Readability of text”.

CONTEXT PROPERTY			
Name	Entity Category	Weight	0,5
Relevance	This attribute can be applied to a number of entity categories.	Multiplicity of Values Measures (value)	0
			Web Application, Web Site, Web Page
Exact Similarity Criterion		Logic Aggregator	OR
		Value Match	Web Application
		Value Match	Web Site
		Value Match	Web Page
CONTEXT PROPERTY			
Name	Main content type	Weight	0,5
Relevance	This attribute can be applied when only text or text and images are the main content of the web site.	Multiplicity of Values Measures (value)	0
Exact Similarity Criterion		Logic Aggregator	OR
		Value Match	Text

Tabla 6.13: Atributo “Identification of content provider”.

(a) Especificación del Atributo “Identification of content provider”.

Name	Identification of Content Provider
Definition	Presence of the identification of the author of a piece of content in a web site, application or page .
Objective	To determine whether the content of the web site is attached to the identification of its author/provider.
PropertyURI	org:IdentificationOfContentProvider
Entity Category	Web Site, Web Application, Web Page

(b) Contexto de aplicación del Atributo “Identification of content provider”.

CONTEXT PROPERTY			
Name	Content Provider Role	Weight	0.33
Relevance	This attribute can be applied when content is provided by users others than the web site owners.	Multiplicity of Values Measures (value)	0
Exact Similarity Criterion		Logic Aggregator	OR
		Value Match	Registered user
CONTEXT PROPERTY			
Name	Entity Category	Weight	0.33
Relevance	This attribute can only by applied to a particular entity category.	Multiplicity of Values Measures (value)	0
Exact Similarity Criterion		Logic Aggregator	OR
		Value Match	Web Application
CONTEXT PROPERTY			
Name	Application Domain	Weight	0.33
Relevance	This attribute can be applied to web application domains where content is provided by users others than the owners.	Multiplicity of Values Measures (value)	0
Exact Similarity Criterion		Logic Aggregator	OR
		Value Match	Social network
		Value Match	File hosting-publishing
		Value Match	Blog

⁵Tomado de <http://thefreedictionary.com/readability>.

También para el concepto *Information Quality>Suitability>Coverage* se encuentra el atributo *Foreign language support* [Calero et al., 2004]⁶, cuya especificación se muestra en la Tabla 6.14a, utilizado para determinar si el sitio web puede ser visto además en idiomas extranjeros (diferentes al idioma del autor/dueño del sitio web). Este atributo se aplica a contextos donde la entidad (categoría) sea un sitio, aplicación o página web, o una aplicación de escritorio tradicional cuya audiencia deseada sea de ámbito internacional (tal como se refleja en la Tabla 6.14b).

Tabla 6.14: Atributo “Foreign Language Support”.

(a) Especificación del Atributo “Foreign language support”.

Name	Foreign Language Support
Definition	Capability of a web site, application or page to be accessed in foreign natural languages.
Objective	To determine if the content of the web site can be viewed in foreign natural languages.
PropertyURI	org:ForeignLanguageSupport
Entity Category	Web Site, Web Application, Web Page

(b) Contexto de aplicación del Atributo “Foreign language support”.

CONTEXT PROPERTY			
Name	Entity Category	Weight	0,5
Relevance	This attribute can only be applied to a number of entity categories.	Multiplicity of Values Measures (value)	0
			Web Application, Web Site, Web Page, Desktop Application
Exact Similarity Criterion		Logic Aggregator	OR
		Value Match	Web Application
		Value Match	Web Site
		Value Match	Web Page
		Value Match	Desktop Application
CONTEXT PROPERTY			
Name	Target Market Scope	Weight	0,5
Relevance	This attribute is applicable when the application is targeted to a regional or international audience.	Multiplicity of Values Measures (value)	0
Exact Similarity Criterion		Logic Aggregator	OR
		Value Match	international

Basado en el cálculo de la métrica *CApp* para cada uno de los atributos vistos arriba –comparando cada contexto de aplicación con el contexto real del proyecto de M&E-, se resuelve incluir al modelo de concepto, bajo los conceptos correspondientes, los atributos *Shopping cart control ease to be recognized*, *Readability of text* y *Foreign language support*, cuyos niveles de aplicabilidad resultante es de 100%; no así en el caso del atributo *Identification of content provider* cuyo nivel de aplicabilidad es de solo 33%.

Así, completando el árbol de requerimientos con los atributos necesarios, se constituye el modelo de concepto que representa los requerimientos no funcionales tal como se muestra en la Figura 6.2.

6.3. Definir e Implementar la Medición

Una vez que se ha completado la especificación de la necesidad de información y los requerimientos no funcionales, se utilizan estos como guía para la definición e implementación de la medición.

6.3.1. Diseño de la medición

Como se indicó en la Sección 4.4, la definición de la medición se basa en la selección de métricas para cuantificar los atributos del modelo de concepto diseñado como parte de los requerimientos no

⁶Definido en dicho trabajo como *Internationalization*.

External Quality

1. Usability
 - 1.1. Understandability
 - 1.1.1. **Shopping cart control ease to be recognized**
 - 1.1.2. **Add-to-cart control ease to be recognized**
 - 1.2. Learnability
 - 1.2.1. **How-to-buy help availability**
 - 1.2.2. **Readability of text**
 - 1.3. Operability
 - 1.3.1. **Shopping cart control permanence**
 - 1.3.2. **Add-to-cart control permanence**
 - 1.3.3. **Steady behavior of the shopping cart control**
 - 1.3.4. **Steady behavior of the Add-to-cart control**
 - 1.4. Accesibility
 - 1.4.1. **Image alternative text availability**
 - 1.4.2. **Support for text-only version**
2. Functional suitability
 - 2.1. Functional completeness
 - 2.1.1. **Capability to add items from anywhere**
 - 2.1.2. **Capability to delete items**
 - 2.1.3. **Capability to modify an item quantity**
 - 2.1.4. **Capability to show totals by performed changes**
 - 2.1.5. **Capability to save items for later/move to cart**
 - 2.2. Functional correctness
 - 2.2.1. **Precision to recalculate after adding an item**
 - 2.2.2. **Precision to recalculate after deleting items**
 - 2.2.3. **Precision to recalculate after modifying an item quantity**
3. Information Quality
 - 3.1. Suitability
 - 3.1.1. Coverage
 - 3.1.1.1. **Foreign language support**
 - 3.1.1.2. **Line item information completeness**
 - 3.1.1.3. **Product description completeness**
 - 3.1.1.4. **Shipping and handling information completeness**
 - 3.1.1.5. **Return policy information completeness**
 - 3.1.2. Consistency
 - 3.1.2.1. **Continue-buying feedback**
 - 3.1.2.2. **Proceed-to-check-out feedback**
 - 3.1.2.3. **User current status feedback**
 - 3.1.3. Value-added
 - 3.1.3.1. **Recently viewed items feedback**
 - 3.1.3.2. **Related items feedback**
 - 3.1.3.3. **Other customers related choices feedback**
 - 3.2. Legal Compliance
 - 3.2.1. **Privacy policy**
 - 3.2.2. **Copyright information**
4. Security
 - 4.1. Confidentiality
 - 4.1.1. **Customer registration**
 - 4.1.2. **Secured check-out**
 - 4.2. Authenticity
 - 4.2.1. **Digital certification of identity**
5. Reliability
 - 5.1. Availability
 - 5.1.1. **Invalid links**
 - 5.1.2. **Check-out service availability**
 - 5.2. Maturity
 - 5.2.1. **Unexpected results dependent of browsers**
 - 5.2.2. **Unexpected results independent of browsers**

Figura 6.2: Modelo de Calidad Externa resultante de la selección de atributos.

funcionales. Como también se describió en el Capítulo 3.2.1 las métricas se obtienen de un *catálogo de M&E* que mantienen especificaciones de métricas previamente definidas, reusables por todos los proyectos de la organización. También se recuerda que, como las métricas son entidades contextuales (según se definió en la Sección 4.4), cada métrica puede tener asociada la descripción de su contexto de aplicación –aquél en el que la métrica puede ser coherentemente utilizada. Se muestra a continuación la selección de la métrica correspondiente para cuatro de los atributos incluidos en el modelo de concepto

instanciado (presentado en la Figura 6.2).

Para el atributo *Readability of text* (código 1.2.2 de la Figura 6.2), cuyo objetivo es determinar la complejidad del texto incluido en la página de ayuda de la aplicación web, se encuentran en el *catálogo de M&E* dos métricas. Una de ellas es la métrica *Fog Index*, cuya especificación se presenta en la Tabla 6.15a. Esta métrica directa permite obtener un valor numérico que representa una estimación de los años de educación formal (según el sistema educativo de EEUU) necesaria para comprender el texto en la primer lectura. La métrica utiliza un método de medición basado en la fórmula de *Robert Gunning*⁷ que efectúa un cálculo en función de la cantidad de palabras, oraciones, y palabras complejas (aquellas con 3 o más sílabas) de un texto. Cabe notar que ésta métrica podría definirse como una métrica indirecta, ya que la fórmula mencionada puede verse como la especificación del método de cálculo correspondiente, y definir métricas directas para cada una de las variables de la fórmula (cantidad de palabras, cantidad de oraciones y cantidad de palabras complejas) dejando que el cálculo de la fórmula sea realizado por la herramienta que implemente el marco C-INCAMI. Sin embargo, se busca reducir la complejidad -innecesaria- inherente a la especificación de dichas métricas, ya que se trata de una fórmula sencilla que puede calcularse fácilmente en un solo paso. Esto demuestra la flexibilidad que ofrece el marco al momento de definir la medición de los atributos del modelo. Por su definición, la métrica *Fog index* es aplicable a textos en idioma inglés y su interpretación se realiza en términos de los niveles de grados o años del sistema educativo de EEUU, según la definición de la escala de la métrica. Estos aspectos contextuales se representan en las dos propiedades incluidas en el contexto de aplicación de la métrica –*Supported Natural Language* y *Website owner's country*–, representado en la Tabla 6.15b.

La segunda métrica que cuantifica el atributo *Readability of text* es *Lexile text* (representada en la Tabla 6.16a). De forma similar a la anterior, ésta métrica representa la dificultad de lectura de un texto mediante una escala numérica expresada en unidad propia –también llamada Lexile. En este caso, el método de medición depende de una herramienta propietaria (*Lexile Analyzer*[®]) que automatiza el cálculo de la métrica, ya que la fórmula para obtener una medida lexile no es publicado por el autor (*MetaMetrics Inc.*). También como la métrica anterior, la métrica *Lexile Text* es considerada una entidad contextual para la cual se describe su *contexto de aplicación* (ver Tabla 6.16b). Éste se describe mediante dos propiedades de contexto. La propiedad *Supported Natural Language* es considerada relevante en este contexto ya que la métrica puede aplicarse sólo para textos en “*inglés*” o “*español*” (recordar que en la Tabla 6.8 se presentó la métrica que cuantifica esta propiedad) por lo que se registran dos mediciones (y medidas correspondientes) asociándolas (*valueMatch*) a un *criterio de similitud exacto*, con un operador de agregación lógica *OR*, para describir los valores alternativos del contexto al cual es aplicable la métrica *Lexile text*. Por otro lado, se especifica la propiedad de contexto *Automated support* para reflejar el requerimiento de contar con la herramienta *Lexile Analyzer*[®] (registrada como valor medido de la propiedad y valor asociado al criterio de similitud) necesaria para obtener valores para el atributo *Readability of text* mediante ésta métrica.

Utilizando el mecanismo de recomendación sensible al contexto, puede verse fácilmente que la métrica que recibe el valor más alto de aplicabilidad contextual es la métrica *Lexile Text* ya que ambas propiedades de su contexto de aplicación satisfacen el criterio de similitud en un 100 % con las propiedades equivalentes en el contexto real del proyecto de M&E (Tabla 6.3), mientras en el contexto de aplicación de la métrica *Fog Index*, ninguna de las dos propiedades satisfacen el criterio de similitud cuando son comparadas con las propiedades equivalentes del contexto real.

Para el atributo *Line item information completeness* (código 3.1.1.2 de la Figura 6.2) se encuentran en el catálogo de M&E dos versiones de una misma métrica, *Degree of completeness to the line item information*. Ambas versiones de la métrica tienen como objetivo comprobar que los ítems en la lista de productos incluidos en el carro de compras contengan todos los datos requeridos para cada tipo de producto, utilizando una escala categórica que permite describir la completitud de los datos esperados

⁷Un empresario norteamericano que desarrolló la fórmula en 1952.

Tabla 6.15: Métrica “Fog Index” para cuantificar el atributo “Readability of Text”.**(a)** Especificación de la Métrica “Fog Index”.

DIRECT METRIC			
Name	Fog Index	Objetive	To obtain the understandability of a text.
References	http://notorc.blogspot.com/2006/09/devils-in-details-measuring.html	Accuracy	100%
Author	Robert Gunning	Version	1.0
Value interpretation	Its output is an approximate representation of the U.S. grade level needed to comprehend a text.		
SCALE (NUMERICAL SCALE)			
Value type	REAL	Scale type	ORDINAL
Representation	CONTINUOUS		
Range Min Value	0.4	Name	Fog Index
Range Max Value	140	Description	An estimation of years of formal USA education needed to understand a text on a first reading.
Step Value	–	Acronym	FI
MEASUREMENT METHOD			
Name	Gunning-FOG formula		
Specification	$0,4 \times \left(\frac{\text{words}}{\text{sentences}} \right) + 100 \times \left(\frac{\text{complex words (w/3+ syllables)}}{\text{words}} \right)$		
References	http://www.idph.state.ia.us/health_literacy/common/pdf/tools/gunning.pdf		

(b) Contexto de aplicación de la métrica “Fog Index”.

CONTEXT PROPERTY			
Name	Supported Natural Language	Weight	0,5
Relevance	This metric can only be applied to a particular natural language.	Multiplicity of Values Measures (value)	1
Exact Similarity Criterion		Logic Aggregator Value Match	OR
			english
CONTEXT PROPERTY			
Name	Website owner’s country	Weight	0,5
Relevance	The interpretation of this metric is based on the Grades Levels used in the USA Education System.	Multiplicity of Values Measures (value)	1
Exact Similarity Criterion		Logic Aggregator Value Match	OR
			USA

en cada caso. Ambas métricas se diferencian en el método de medición que enumera los datos esperados para los diferentes tipos de productos comercializados a través de la aplicación web. La versión 1.0, presentada en la Tabla 6.17a, define la especificación de su método de medición requiriendo la presencia de datos característicos de libros y CDs de música, mientras el método de medición de la versión 2.0 (presentada en la Tabla 6.18a) sólo comprueba la presencia de datos sobre libros. Ésta particularidad es reflejada en el contexto de aplicación de cada métrica mediante la propiedad de contexto *Traded product type*. Para la versión 1.0, esta propiedad de contexto toma los valores “books” y “music” (ver Tabla 6.17b), mientras que en la versión 2.0 sólo se incluye el valor “books” para dicha propiedad de contexto (ver Tabla 6.18b). Adicionalmente, el contexto de aplicación de ambas métricas incluyen las propiedades *Entity Category* y *Application Domain*, para reflejar en ambos casos que la métrica correspondiente es aplicable a “Aplicaciones Web” en el dominio de “e-commerce” –respectivamente–, en las cuales existe un carro de compras en el que el usuario cliente puede agregar ítems que luego podrán ser efectivamente comprados.

Utilizando el mecanismo de recomendación sensible al contexto para ambas métricas, puede observarse que, en la comparación con el contexto real, el criterio de similitud de las propiedades *Entity Category* y *Application Domain* son satisfechas en un 100 % para ambas métricas. Contrariamente, el criterio de similitud de la propiedad *Traded Product Type* en el contexto de aplicación de la métrica v1.0 requiere que se encuentren los valores “books” y “music” en el contexto real, cuando éste último sólo incluye el valor “books” para dicha propiedad. Por el contrario, en el contexto de aplicación de la versión 2.0 de la métrica, el criterio de similitud de la misma propiedad sólo requiere la presencia

Tabla 6.16: Métrica “Lexile Text” para cuantificar el atributo “Readability of Text”.

(a) Especificación de la Métrica “Lexile Text”.

DIRECT METRIC			
Name	Lexile Text	Objetivo	To obtain a numeric representation of a text’s readability.
References	http://www.lexile.com/about-lexile/lexile-overview/	Accuracy	100 %
Author	MetaMetrics, Inc.	Version	1.0
Value interpretation	The numeric representation of a text’s readability (or difficulty). The bigger the value, the higher the level of difficulty of the text.		
SCALE (NUMERICAL SCALE)			
Value type	INTEGER	Scale type	ABSOLUTE
Representation	DISCRETE	UNIT	
Range Min Value	0	Name	Lexile
Range Max Value	2000	Description	The numeric representation of an individual’s reading ability or a text’s readability (or difficulty)
Step Value	10	Acronym	L
MEASUREMENT METHOD			
Name	Lexile Text Measure		
Specification	Follow the Lexile Analyzer®Guidelines for analyzing texts.		
References	http://www.lexile.com/tools/lexile-analyzer/using-the-professional-analyzer/		
Automated By	SOFTWARE TOOL		
	Name	Lexile Analyzer®	
	Description	A software program that evaluates the reading demand–or readability–of books, articles and other materials. The Lexile Analyzer measures the complexity of the text by breaking down the entire piece and studying its characteristics, such as sentence length and word frequency, which represent the syntactic and semantic challenges that the text presents to a reader. The outcome is the text complexity, expressed as a Lexile measure, along with information on the word count, mean sentence length and mean log frequency.	
	Provider	MetaMetrics	
	Version	–	

(b) Contexto de aplicación de la métrica “Lexile Text”.

CONTEXT PROPERTY			
Name	Supported Natural Language	Weight	0,5
Relevance	This metric can be applied to a number of natural languages.	Multiplicity of Values	0
		Measures (value)	english, spanish
Exact Similarity Criterion		Logic Aggregator	OR
		Value Match	english
		Value Match	spanish
CONTEXT PROPERTY			
Name	Automated support	Weight	0,5
Relevance	This metric requires the use of a particular tool to perform the corresponding measurement.	Multiplicity of Values	1
		Measures (value)	Lexile Analyzer®
Exact Similarity Criterion		Logic Aggregator	OR
		Value Match	Lexile Analyzer®

del valor “books”. Por lo tanto, la métrica más apropiada, y en consecuencia la seleccionada para cuantificar el atributo *Line item information completeness*, es la versión 2.0 de *Degree of completeness to the line item information*.

De forma similar al atributo *Line item information completeness*, el atributo *Product description completeness* (código 3.1.1.3 de la Figura 6.2) tiene como objetivo determinar que en la descripción de los productos comercializados por la aplicación web de e-commerce se encuentren todos los datos requeridos del producto, que incluyen todos los requeridos por la métrica asociada al atributo *Product description completeness* más otros adicionales asociados a cada tipo de producto (por ejemplo, para un CD de música debe incluirse la lista de temas; para un libro se debe incluir la tabla de contenidos). Para este segundo atributo, se aplican las mismas propiedades de contexto de aplicación; particularmente, los valores de la propiedad *Traded Product Type* deben incluir los tipos de producto cuyos datos son

Tabla 6.17: Métrica “*Degree of completeness to the line item information*” (v1.0) para cuantificar el atributo “*Line item information completeness*”.

(a) Especificación de la Métrica “*Degree of completeness to the line item information*” (v1.0).

DIRECT METRIC			
Name	Degree of completeness to the line item information	Objetive	Check for the presence of all data corresponding to the item product type.
References	Olsina <i>et al.</i> [2008a]	Accuracy	100 %
Author	Luis Olsina	Version	1.0
Value interpretation	A categorical value indicating whether the line item information includes the expected data fields.		
SCALE (CATEGORICAL SCALE)			
Value type	SYMBOL	Scale type	ORDINAL
Allowed values	[incomplete partially complete complete]		
MEASUREMENT METHOD			
Name	Determination of line item information completeness		
Specification	Include in the shopping cart a product of each type traded by the e-commerce web site and determine for all of them if the following data fields are present in the line item: (for all kind of products) title, price, quantity, added on date, availability, (for music products) artist, label, style, release date, (for books products) author, editor, isbn. It is <i>incomplete</i> if all data related to a particular product type is missing for at least one product type; it is <i>partially complete</i> if some data related to a particular product type is missing for at least one product type; it is <i>complete</i> if all expected data is present for all product types.		
References	http://www.lexile.com/tools/lexile-analyzer/using-the-professional-analyzer/		

(b) Contexto de aplicación de la métrica “*Degree of completeness to the line item information*” (v1.0).

CONTEXT PROPERTY			
Name	Traded product type	Weight	0.33
Relevance	The measurement method of this metric is defined in terms of the properties of certain kind	Multiplicity of Values	0
		Measures (value)	books, music
Exact Similarity Criterion		Logic Aggregator	XAND
		Value Match	books
		Value Match	music
CONTEXT PROPERTY			
Name	Entity Category	Weight	0.33
Relevance	This metric can only be applied to a particular entity category.	Multiplicity of Values	0
		Measures (value)	Web Application
Exact Similarity Criterion		Logic Aggregator	OR
		Value Match	Web Application
CONTEXT PROPERTY			
Name	Application Domain	Weight	0,5
Relevance	This metric can only be applied to e-commerce applications where a shopping cart component is present including line items for products in the cart.	Multiplicity of Values	0
		Measures (value)	e-commerce
Exact Similarity Criterion		Logic Aggregator	OR
		Value Match	e-commerce

requeridos en la especificación del método de medición de la métrica que cuantifica al atributo.

Para el atributo *Shipping and handling information completeness* (código 3.1.1.4 de la Figura 6.2) se encuentra en el catálogo de M&E una única métrica que lo cuantifica: *Available shipping and handling information* (ver Tabla 6.19). Esta métrica comprueba que se indique costo y tiempo de entrega para todos los posibles destinos donde se distribuyen los productos, expresándolo en una escala categórica que describe. El método de medición indica que el valor de la medición se determina manualmente comprobando que dicha información se encuentre presente en el sitio web.

Aunque ésta es la única métrica encontrada para cuantificar el atributo mencionado, el contexto de aplicación debe ser igualmente considerado ya que podría no ser aplicable al contexto real del proyecto, en cuyo caso se debiera diseñar una métrica que sí lo hiciera. En este caso, el contexto de aplicación de la métrica se describe mediante dos propiedades, *Entity Category* y *Application Domain*, con los valores “*Web Application*” y “*e-commerce*” respectivamente (la especificación de estas propiedades no es descripta en una tabla ya que son similares a otras presentadas anteriormente). Ambas propiedades

Tabla 6.18: Métrica “*Degree of completeness to the line item information*” (v2.0) para cuantificar el atributo “*Line item information completeness*”.(a) Especificación de la Métrica “*Degree of completeness to the line item information*” (v2.0).

DIRECT METRIC			
Name	Degree of completeness to the line item information	Objetive	Check for the presence of all data corresponding to the item product type.
References	Olsina <i>et al.</i> [2008a]	Accuracy	100 %
Author	Luis Olsina	Version	2.0
Value interpretation	A categorical value indicating whether the line item information includes the expected data fields.		
SCALE (CATEGORICAL SCALE)			
Value type	SYMBOL	Scale type	ORDINAL
Allowed values	[incomplete partially complete complete]		
MEASUREMENT METHOD			
Name	Determination of line item information completeness		
Specification	Include in the shopping cart a product of each type traded by the e-commerce web site and determine for all of them if the following data fields are present in the line item: (for all kind of products) title, price, quantity, added on date, availability, (for books products) author, editor, isbn. It is <i>incomplete</i> if all data related to a particular product type is missing for at least one product type; it is <i>partially complete</i> if some data related to a particular product type is missing for at least one product type; it is <i>complete</i> if all expected data is present for all product types.		
References	http://www.lexile.com/tools/lexile-analyzer/using-the-professional-analyzer/		

(b) Contexto de aplicación de la métrica “*Degree of completeness to the line item information*” (v2.0).

CONTEXT PROPERTY			
Name	Traded product type	Weight	0.33
Relevance	The measurement method of this metric is defined in terms of the properties of certain kind	Multiplicity of Values Measures (value)	0 books
Exact Similarity Criterion		Logic Aggregator Value Match	XAND books
CONTEXT PROPERTY			
Name	Entity Category	Weight	0.33
Relevance	This metric can only be applied to a particular entity category.	Multiplicity of Values Measures (value)	0 Web Application
Exact Similarity Criterion		Logic Aggregator Value Match	OR Web Application
CONTEXT PROPERTY			
Name	Application Domain	Weight	0,5
Relevance	This metric can only be applied to e-commerce applications where a shopping cart component is present including line items for products in the cart.	Multiplicity of Values Measures (value)	0 e-commerce
Exact Similarity Criterion		Logic Aggregator Value Match	OR e-commerce

son consideradas relevantes porque describen el tipo de entidad para la cual se espera que exista información de costo y tiempo de envío de productos comprados on-line. Como se puede observar, los valores de ambas propiedades satisfacen al 100 % los criterios de similitud correspondientes cuando se comparan con el contexto real del proyecto de M&E. Por lo tanto la métrica es asignada al atributo *Shipping and handling information completeness*.

Para el atributo *Foreign Language Support* se encuentran dos métricas. La primera de ellas es *Supported Natural Languages* (representada en la Tabla 6.20) cuya escala categórica describe los nombres de los idiomas que pueden ser registrados por la métrica. El método de medición de esta métrica indica registrar los nombres de los idiomas en los que efectivamente puede verse el contenido del sitio al activar las opciones del control correspondiente. La segunda métrica, llamada *Amount of Supported Natural Languages* (ver Tabla 6.21) utiliza, contrariamente a la anterior, una escala numérica que registra la cantidad de idiomas en los que se puede acceder el contenido del sitio web. El método de medición en este caso es similar al anterior sólo que se registra la cantidad de idiomas a los que efectivamente puede cambiarse el texto del sitio. Para ambas métricas, el contexto de aplicación es el mismo, incluyendo sólo la propiedad de contexto *Entity Category* con los valores “*Web application*”,

Tabla 6.19: Especificación de la Métrica “*Available shipping and handling information*” para cuantificar el atributo “*Shipping and handling information completeness*”.

DIRECT METRIC			
Name	Available shipping and handling information	Objetive	To check for the presence of all data corresponding to shipping and handling costs and time associated to the delivery of goods sold in the web site to the postal address of the buyer.
References	Olsina <i>et al.</i> [2008a]	Accuracy	100 %
Author	Luis Olsina	Version	1.0
Value interpretation	A categorical value indicating whether shipping and handling costs information is available for all possible destinies.		
SCALE (CATEGORICAL SCALE)			
Value type	SYMBOL	Scale type	ORDINAL
Allowed values	[shipping and handling cost and time are indicated for all destinies shipping and handling cost and time are indicated only for some destinies one of shipping and handling cost or time is missing both shipping and handling cost and time are missing]		
MEASUREMENT METHOD			
Name	Determination of shipping and handling information		
Specification	Check for the presence within the web site of information regarding shipping and handling cost and time for all destinies to which products can be delivered by the company of the e-commerce web application.		
References	-		

Tabla 6.20: Métrica “*Supported Natural Languages*” para cuantificar el atributo “*Foreign Language Support*”.

DIRECT METRIC			
Name	Supported Natural Languages	Objetive	To determine the natural languages in which the web application can be accessed/viewed.
References	...	Accuracy	100 %
Author	...	Version	1.0
Value interpretation	Categorical values identifying the natural languages in which the web application can be accessed/viewed.		
SCALE (CATEGORICAL SCALE)			
Value type	SYMBOL	Scale type	ORDINAL
Allowed values	[spanish english portuguese italian french german]		
MEASUREMENT METHOD			
Name	Determination of supported natural languages		
Specification	Change the language in which the web application is seen by using the language control included in it. Repeat for each language included in the control options. Record the names of all the languages for which the natural language of the web site content changed effectively.		
References	-		

Tabla 6.21: Métrica “*Amount of Supported Natural Languages*” para cuantificar el atributo “*Foreign Language Support*”.

DIRECT METRIC			
Name	Amount of Supported Natural Languages	Objetive	To determine the amount of natural languages in which the web application can be accessed/viewed.
References	...	Accuracy	100 %
Author	...	Version	1.0
Value interpretation	The amount of natural languages in which the web application can be accessed/viewed.		
SCALE (NUMERICAL SCALE)			
Value type	INTEGER	Scale type	ABSOLUTE
Representation	DISCRETE	UNIT	
Range Min Value	0	Name	Natural Language
Range Max Value	infinite	Description	Amount of natural languages.
Step Value	-	Acronym	NL
MEASUREMENT METHOD			
Name	Count of supported natural languages		
Specification	Count one natural language for each language included in the language control options of the web site for which the natural language of the web site content changes effectively when the corresponding option is activated.		
References	-		

“Web page”, “Web Site” y “Desktop application” (se obvia la especificación de los contextos por su simplicidad). En este caso la decisión sobre qué métrica utilizar se realiza mediante el juicio del evaluador que destaca la necesidad de conocer qué lenguajes son soportados por el sitio, ya que la cantidad de estos no aporta ninguna información útil para la evaluación.

La mayoría de las métricas para los atributos restantes del modelo de concepto de la Figura 6.2 son aplicables a un contexto similar al descrito para la métrica *Available shipping and handling information* descrita más arriba (ver Tabla 6.19), es decir incluyendo las propiedades *Entity Category* y *Application Domain*, con los valores “Web Application” y “e-commerce” respectivamente.

6.3.2. Implementación de la medición

Una vez que se ha asignado una métrica a cada uno de los atributos del modelo de concepto de la Figura 6.2, es posible llevar a cabo la implementación de la medición. Como se describió en el Capítulo 4, ésta consiste en efectuar las mediciones para todos los atributos del modelo siguiendo la especificación de cada métrica asociada, y registrando los resultados utilizando los conceptos provistos por el modelo, es decir objetos *Measurement* asociados a la entidad sobre la cual se realiza la medición.

En la Tabla 6.22 se presentan los resultados de las mediciones para las métricas descritas en la subsección anterior (se omite el atributo *collectorContactInfo* para las mediciones incluidas en la tabla).

Tabla 6.22: Mediciones realizadas sobre la entidad *Cuspide.com* para las métricas vistas en la Subsección 6.3.1.

Metric	Data collector	Timestamp	Value (Measure)
Lexile Text	Hernán Molina	2008-11-19 16:13	750L
Degree of completeness to the line item information” (v2.0)	Hernán Molina	2008-11-19 17:34	partially complete
Available shipping and handling information	Pablo Becker	2008-11-20 10:08	shipping and handling cost and time are indicated for all destinies
Supported Natural Languages	Pablo Becker	2008-11-20 10:20	spanish

6.4. Definir e Implementar la Evaluación

Según se describió en la Sección 4.5, la evaluación se diseña seleccionando *modelos elementales* y *globales* que especifiquen cómo se convierten los valores medidos, provenientes de las diversas escalas heterogéneas de las métricas utilizadas, a valores de preferencia normalizados a una escala que permita interpretar los valores medidos para todos los niveles del modelo de concepto de manera homogénea, utilizando criterios de decisión previamente acordados. Al igual que en los casos anteriores, los *modelos elementales*, *globales* y *criterios de decisión* son *entidades contextuales* cuya aplicación apropiada puede depender del contexto en el cual pretenden utilizarse. En este caso se utiliza nuevamente el mecanismo de recomendación sensible al contexto propuesto en esta tesis para dar soporte a tales decisiones.

Antes de continuar con este aspecto, cabe resaltar lo siguiente. Ya que los indicadores elementales y globales pretenden homogeneizar la escala de los valores que describen los niveles de preferencia para cada componente del modelo de concepto, la elección de los modelos elementales y globales debería ser realizada en función de dicha escala. Esta puede ser seleccionada previamente al diseño de la evaluación elemental y global –en una etapa previa de diseño de criterios comunes- o puede ser determinada de forma indirecta luego de la selección de un modelo global o elemental, ya que estos últimos determinan y/o dependen implícitamente de una escala en particular. En cualquier caso, se podría considerar que la escala seleccionada establece una propiedad más del contexto real del proyecto, en cuyo caso, los modelos elementales y globales podrían especificar tal propiedad como parte de sus contextos de aplicación indicando la escala de la cual dependen –el dominio- y, asimismo, la escala correspondiente al valor de salida –el rango- para ser utilizada en la selección de los modelos más apropiados. No obstante,

y teniendo en cuenta que estas definiciones se llevan a cabo en el entorno de una misma organización, es lícito considerar que, acudiendo a un criterio de coherencia y consistencia de la gerencia, la escala usada por todos los modelos elementales y globales definidos en el catálogo de M&E de la organización sea una constante. De allí que incluir la escala como una propiedad del contexto de aplicación de modelos elementales y globales sería una redundancia. Sin embargo, no se descarta que la naturaleza de la organización requiera contar con diferentes escalas para la valuación de requerimientos no funcionales. En tal caso, es posible implementar el enfoque planteado al principio de este párrafo.

En la prueba de concepto del presente capítulo se ilustra el caso donde la organización utiliza de forma general para todos sus proyectos una única escala para todos los valores de preferencia de requerimientos no funcionales. Se trata en este caso de una *escala numérica*, de tipo *RATIO*, de representación *CONTINUOUS* y valores de tipo *REAL* que se encuentran en el rango entre 0,0 y 1,0.

6.4.1. Selección de modelos elementales

A continuación se debe seleccionar un *modelo elemental* (*ElementaryModel*) para cada *indicador elemental* (*ElementaryIndicator*) que interpreta un atributo del modelo de concepto de la Figura 6.2, y que se utilizará para convertir valores de la escala de la métrica, que cuantifica a cada atributo, a valores de preferencia en una escala normalizada. Se recuerda (como en los casos anteriores) que los modelos elementales son considerados entidades contextuales (ver Figura 4.7) por lo que, para los casos en los que se describa el contexto de aplicación, puede utilizarse el mecanismo de recomendación sensible al contexto para dar soporte a la selección del modelo más apropiado.

A continuación se describe la selección de modelos elementales para tres de los atributos para los cuales se ilustró la asignación de métricas en la Subsección 6.3.1. En cada caso, se consideran los modelos elementales que interpretan la métrica que cuantifica el atributo en cuestión. Un modelo elemental se especifica, como se vio en la Sección 4.5, básicamente mediante un nombre y su especificación, que consiste en una fórmula o ecuación para llevar a cabo la conversión mencionada. En la especificación de los modelos elementales que se ilustran a continuación, se utiliza el término M_{XY} para referirse a la medida –o medidas (valores)- registradas a partir de la métrica XY .

Para interpretar el atributo *Foreign Language Support* (código 3.1.1.1), cuantificado por la métrica *Supported Natural Languages* (presentada en la Tabla 6.20), se encuentran tres posibles *modelos elementales* en el catálogo de M&E. El primero de los modelos elementales encontrados es el EM_{SNL1} , cuya especificación se muestra en la Ecuación 6.1, en la cual puede observarse que el atributo correspondiente es interpretado en función de los idiomas “*spanish*” y “*portuguese*”. El modelo comprueba que los valores esperados se encuentren entre aquellos medidos por la métrica *Supported Natural Languages*. A diferencia de los modelos vistos arriba, en este modelo se contemplan todas las medidas (valores) registradas a partir de la métrica correspondiente, utilizando el operador \in (pertenencia) para determinar si un determinado valor se encuentra entre ellas.

$$EM_{SNL1} = \begin{cases} 0 & \text{si "spanish" } \in M_{SNL} \\ 100 & \text{si "spanish" } \in M_{SNL} \wedge \text{"portuguese" } \in M_{SNL} \end{cases} \quad (6.1)$$

Este modelo es aplicable a contextos en los cuales la *región geográfica objetivo* del sitio web es “*South America*” y el *país del propietario del sitio web* sea “*Argentina*” (ver Tabla 6.23). De esta forma cuando sólo el lenguaje correspondiente al *país del propietario del sitio web* se encuentra disponible como opción en el sitio, el modelo elemental asigna un valor de preferencia 0% ya que no estaría soportando ningún idioma extranjero. Asimismo, sólo los idiomas encontrados en tal región geográfica son incluidos en la ecuación, de manera que si todos ellos se encuentran entre los valores medidos por la métrica, el atributo será evaluado al 100%.

Tabla 6.23: Contexto de aplicación del Modelo Elemental EM_{SNL1} que evalúa al atributo *Foreign Language Support*.

CONTEXT PROPERTY			
Name	Target Geographic Region	Weight	0,5
Relevance	The specification of this elementary model assumes the web site has a particular target geographic region because of the natural languages included in it.	Multiplicity of Values Measures (value)	1
			South America
Exact Similarity Criterion		Logic Aggregator	XAND
		Value Match	South america
CONTEXT PROPERTY			
Name	Web Site owner's country	Weight	0,5
Relevance	The specification of this elementary model assumes a particular home country of the web site its natural language is not considered a foreign language.	Multiplicity of Values Measures (value)	1
			Argentina
Exact Similarity Criterion		Logic Aggregator	OR
		Value Match	Argentina

El segundo modelo elemental encontrado es el EM_{SNL2} , cuya especificación se representa en la Ecuación 6.2. A diferencia del anterior, en este modelo el atributo es evaluado en función de los idiomas *español*, *portugués* e *inglés*, asignando un valor de 0% cuando el único idioma encontrado es el *español*. Según dicha especificación, este modelo es aplicable en contextos donde la *región geográfica objetivo* incluye “*South America*” y “*North America*” y donde el *país del propietario del sitio web* es “*Argentina*” (tal como se representa en la Tabla 6.24).

$$EM_{SNL2} = \begin{cases} 0 & \text{si "spanish" } \in M_{SNL} \\ 70 & \text{si "spanish" } \in M_{SNL} \wedge \text{"portuguese" } \in M_{SNL} \\ 100 & \text{si "spanish" } \in M_{SNL} \wedge \text{"portuguese" } \in M_{SNL} \wedge \text{"english" } \in M_{SNL} \end{cases} \quad (6.2)$$

Tabla 6.24: Contexto de aplicación del Modelo Elemental EM_{SNL2} que evalúa al atributo *Foreign Language Support*.

CONTEXT PROPERTY			
Name	Target Geographic Region	Weight	0,5
Relevance	The specification of this elementary model assumes the web site has a particular target geographic region because of the natural languages included in it.	Multiplicity of Values Measures (value)	1
			South America, North America
Exact Similarity Criterion		Logic Aggregator	XAND
		Value Match	South America North America
CONTEXT PROPERTY			
Name	Web Site owner's country	Weight	0,5
Relevance	The specification of this elementary model assumes a particular home country of the web site. Its natural language is not considered a foreign language.	Multiplicity of Values Measures (value)	1
			Argentina
Exact Similarity Criterion		Logic Aggregator	OR
		Value Match	Argentina

Finalmente, el tercer modelo elemental encontrado es el EM_{SNL3} , cuya especificación se representa en la Ecuación 6.3. En este modelo el atributo es evaluado en función de los idiomas *español*, *portugués*, *inglés*, *italiano*, *francés* y *alemán*, asignando un valor de 0% cuando el único idioma encontrado es el *español*. Debido a los idiomas incluidos en la especificación del modelo, éste es aplicable en contextos donde la *región geográfica objetivo* incluye “*South America*”, “*North America*”, “*Europe*”, “*Asia*”, “*Africa*” y “*Oceania*” y donde el *país del propietario del sitio web* es “*Argentina*” (tal como se representa en la Tabla 6.25).

$$EM_{SNL3} = \begin{cases} 0 & \text{si "spanish"} \in M_{SNL} \\ 30 & \text{si "spanish"} \in M_{SNL} \wedge \text{"portuguese"} \in M_{SNL} \\ 70 & \text{si "spanish"} \in M_{SNL} \wedge \text{"portuguese"} \in M_{SNL} \wedge \text{"english"} \in M_{SNL} \\ 100 & \text{si "spanish"} \in M_{SNL} \wedge \text{"portuguese"} \in M_{SNL} \wedge \text{"english"} \in M_{SNL} \\ & \wedge \text{"italian"} \in M_{SNL} \wedge \text{"french"} \in M_{SNL} \wedge \text{"german"} \in M_{SNL} \end{cases} \quad (6.3)$$

Tabla 6.25: Contexto de aplicación del Modelo Elemental EM_{SNL3} que evalúa al atributo *Foreign Language Support*.

CONTEXT PROPERTY			
Name	Target Geographic Region	Weight	0,5
Relevance	The specification of this elementary model assumes the web site has a particular target geographic region because of the natural languages included in it.	Multiplicity of Values Measures (value)	1 South America, North America, Europe, Asia, Africa, Oceania
Exact Similarity Criterion		Logic Aggregator	XAND
		Value Match	South America
		Value Match	North America
		Value Match	Europe
		Value Match	Asia
		Value Match	Africa
		Value Match	Oceania
CONTEXT PROPERTY			
Name	Web Site owner's country	Weight	0,5
Relevance	The specification of this elementary model assumes a particular home country of the web site its natural language is not considered a foreign language.	Multiplicity of Values Measures (value)	1 Argentina
Exact Similarity Criterion		Logic Aggregator	OR
		Value Match	Argentina

Recurriendo al mecanismo de recomendación sensible al contexto, se determina la aplicabilidad contextual de cada uno de los modelos elementales al contexto real del proyecto de M&E. En primer lugar, se puede observar que, para los tres modelos, el criterio de similitud de la propiedad *Web Site owner's country* en los contextos de aplicación correspondientes es satisfecho en un 100%. Por lo tanto, la aplicabilidad final para los tres modelos se determina mediante la propiedad *Target Geographic Region* que, en los tres casos, especifica un *criterio de similitud exacto* con un operador XAND –todos y únicamente los valores registrados por el criterio son esperados en el contexto real. Observando los valores especificados para dicha propiedad en el contexto real del proyecto de M&E (Tabla 6.3), se puede ver que los criterios de similitud de dicha propiedad en los contextos de aplicación de los modelos EM_{SNL1} y EM_{SNL2} no son satisfechos, resultando en un nivel de aplicabilidad contextual $CApp$ de 50% en ambos casos; por lo contrario, el criterio de similitud de la propiedad equivalente del contexto de aplicación del modelo EM_{SNL3} especifica los mismos valores (y cantidad de los mismos) que en el contexto real, resultando finalmente en un nivel de aplicabilidad contextual $CApp$ de 100%, por lo que éste último modelo es elegido para ser asignado al indicador que interpretará al atributo *Foreign Language Support*.

Para el atributo *Readability of Text* (código 1.2.2 de la Figura 6.2), cuantificado por la métrica *Lexile Text* (presentada en la Tabla 6.16), se encuentra un único modelo elemental que convierte los valores de la escala de la métrica a tres posibles valores de indicador (ver Ecuación 6.4). Este modelo interpreta con un valor de indicador 100% a valores de la métrica que sean menores o iguales a 800, mientras valores mayores hasta 1500 son interpretados como 50%; en otro caso el valor de la métrica se interpreta como 0%. Se debe recordar, como se expuso en la Tabla 6.16, que la métrica *Lexile Text* presenta valores mayores cuanto mayor es la dificultad de leer un texto.

$$EM_{LT} = \begin{cases} 0 & \text{si } 1500 < M_{LT} \\ 50 & \text{si } 800 < M_{LT} \leq 1500 \\ 100 & \text{si } 0 \leq M_{LT} \leq 800 \end{cases} \quad (6.4)$$

Este modelo es aplicable a un contexto en el cual la *categoría de entidad* sea una *página web*, un *sitio web* o una *aplicación web* (por la simplicidad de la descripción de dicho contexto, se omite su especificación). Además de que éste modelo es el único encontrado para interpretar la métrica *Lexile Text*, el contexto real (Tabla 6.3) satisface el criterio de similitud correspondiente –un criterio de similitud exacto con un operador OR- por lo que es seleccionado para evaluar el atributo *Readability of Text*.

Para el atributo *Line item information completeness* (código 3.1.1.2 de la Figura 6.2), cuantificado por la métrica *Degree of completeness to the line item information (v2.0)* (presentada en la Tabla 6.18) se encuentra también un único modelo elemental que convierte los valores de la escala categórica de la métrica a valores de la escala del indicador tal como se describe en la Ecuación 6.5.

$$EM_{DCLIV2} = \begin{cases} 0 & \text{si } M_{DCLIV2} = \text{"incomplete"} \\ 50 & \text{si } M_{DCLIV2} = \text{"partially complete"} \\ 100 & \text{si } M_{DCLIV2} = \text{"complete"} \end{cases} \quad (6.5)$$

Este modelo es aplicable a contextos donde la *categoría de entidad* sea una *página web*, un *sitio web* o una *aplicación web* y donde el *dominio de aplicación* sea *e-commerce* (ya que el modelo interpreta una métrica aplicable a ese mismo contexto). Ambas propiedades de contexto tienen un *criterio de similitud exacto* con un operador OR. El contexto real del proyecto (Tabla 6.3) satisface el criterio de similitud de las propiedades contexto de aplicación en un 100 % por lo que el modelo es asignado al indicador elemental que evalúa el atributo *Line item information completeness*.

6.4.2. Selección de un modelo global

Una vez que se han asignado modelos elementales a cada atributo del modelo de concepto, se debe decidir un *modelo global* (*GlobalModel*) para cada indicador parcial/global que especifique el mecanismo de agregación de los valores de preferencia elemental (proveniente de indicadores elementales) y parcial (proveniente de indicadores globales de nivel intermedio) tal como se especifica en el módulo *evaluation* (representado en la Figura 4.7). Como se indicó en el Capítulo 4, la aplicación de un modelo global depende principalmente de la disponibilidad de los recursos de cómputo necesarios para llevar a cabo el método de cálculo asociado, aspecto observable en el contexto real del proyecto de M&E.

En el catálogo de M&E se encuentran dos modelos globales. El primero de ellos es un modelo lineal aditivo que toma como entrada un número de valores de preferencia elementales o parciales ($PE \setminus P_i$) y pesos relativos (W_i) para cada uno de ellos, y realiza la agregación en la forma de una suma ponderada, de la siguiente forma (Ecuación 6.6):

$$PP \setminus G = (W_1 PE \setminus P_1 + W_2 PE \setminus P_2 + \dots + W_m PE \setminus P_m) \quad (6.6)$$

de manera que, si los valores de preferencia $PE \setminus P_i$ cumplen que $0 \leq PE \setminus P_i \leq 1$ y la suma de los pesos W_i cumple que $(W_1 + W_2 + \dots + W_m) = 1$; si $W_i > 0$; para $i = 1 \dots m$ entonces la ecuación 6.6 satisface $0 \leq PP \setminus G \leq 1$.

Como se mencionó más arriba, un modelo global es considerado una entidad contextual. Este modelo en particular es aplicable en contextos donde se disponga del soporte tecnológico para el cálculo de funciones lineales aditivas. Tal es el caso del contexto real del proyecto de M&E, como se observa en la propiedad *Automated support* de la Tabla 6.3, implementado por la misma herramienta que da soporte tecnológico al marco C-INCAMI [Olsina *et al.*, 2008b].

El otro modelo global disponible en el catálogo de M&E es el *Modelo LSP (Logic Scoring of Preference)* [Dujmovic, 1996], que calcula el valor de preferencia de cada indicador global tomando como entrada un número de valores de preferencia elementales o parciales (e_i), pesos relativos (W_i) para cada uno de ellos y un operador r que asigna un comportamiento lógico para todos los valores de preferencia que opera en un rango que va desde la conjunción pura a la disyunción pura (ver Ecuación 6.7).

$$e_0 = (W_1 e_1^r + \dots + W_k e_k^r)^{1/r}, \quad (6.7)$$

donde $(W_1 + \dots + W_k) = 1, W_i > 0, i = 1, \dots, k$.

De la misma forma que el modelo lineal, el modelo LSP es aplicable en contextos en los que se disponga del soporte tecnológico correspondiente –contiene la propiedad de contexto *Automated Support* con el valor “*LSP Method*”. Nuevamente, el contexto real del proyecto de M&E satisface dicho requerimiento.

Por lo anterior, ambos modelos globales son contextualmente aplicables al proyecto de M&E por lo que la decisión de qué modelo global utilizar se realiza según el juicio del evaluador que prioriza la posibilidad de especificar diferentes niveles de simultaneidad o conjunción para los indicadores correspondientes a los distintos componentes del modelo de concepto. Por lo tanto se selecciona el Modelo LSP como modelo global para todos los indicadores globales que interpretan los conceptos del modelo de concepto de la Figura 6.2.

6.4.3. Configuración de los indicadores globales y elementales

Una vez que se ha seleccionado el modelo global que define el mecanismo de agregación de preferencias elementales y parciales, es necesario configurar tanto los indicadores globales, como así también los indicadores elementales, de acuerdo a las particularidades del modelo elegido –en este caso, el modelo LSP [Dujmovic, 1996]. Según se describió en la Subsección 3.2.3, el método LSP establece que se deben decidir los operadores a utilizar en cada función de agregación correspondiente a cada indicador global (ilustrados en la Figura 3.4), así como los pesos relativos de cada uno de los componentes/entradas a dichas funciones, ya sean indicadores elementales o parciales –indicadores globales de nivel medio. Estos parámetros son registrados utilizando los metadatos del módulo *evaluation* del marco C-INCAMI (mediante el atributo *weight* del concepto *Indicator* –y por lo tanto común a ambos tipos de indicadores– y el atributo *operator* del concepto *GlobalIndicator*).

En este caso el evaluador recurre al conocimiento de expertos del dominio para determinar los operadores y los pesos que serán asignados a los componentes de cada función. En la Figura 6.3 se representa parte del árbol de indicadores globales y elementales que interpretan el modelo de concepto para *External Quality*, mostrando para cada indicador el peso y operador (cuando corresponda) asignados. Los indicadores globales son representados con el prefijo (GI) –por *Global Indicator*– mientras que los indicadores elementales se representan con la prefijo (EI) –de *Elementary Indicator*.

Para el concepto de alto nivel *External Quality* se asigna un operador de simultaneidad, particularmente el operador de *cuasiconjunción* de nivel medio *CA*, utilizado para modelar requerimientos obligatorios. De esta forma, si el cálculo de la función de preferencia de alguno de los conceptos que

componen *External Quality* arroja 0 como resultado –es decir que el requerimiento no es satisfecho en absoluto- tal será el resultado de satisfacción para dicho concepto. De forma similar, un valor bajo de preferencia en alguno de los subconceptos de *External Quality* penalizará el resultado final según el grado de conjunción correspondiente al operador *CA*. Los pesos para los componentes de la función de agregación para este indicador son asignados más o menos de forma balanceada (tal como puede observarse en la figura citada arriba), dando mayor importancia relativa al concepto *Information Quality*, que recibe un peso de 0,3, siguiéndole los conceptos *Usability*, *Functional Suitability* y *Security*, con un peso de 0,2, y finalmente *Reliability* con un peso de 0,1 (cabe recordar que, según define el método LSP, los pesos de los componentes de la función de agregación deben sumar 1,0). Para la función de agregación correspondiente a los indicadores globales de estos últimos conceptos se utiliza el operador *C–*, también representando una relación de simultaneidad (aunque en este caso modela requerimientos no obligatorios, como se vio en la sección 3.2.3.1), a excepción del concepto *Security* que utiliza un operador *CA*, ya que los conceptos que lo componen –*Confidentiality* y *Authenticity*- se consideran obligatorios según los requerimientos determinados por el evaluador.

- (GI) External Quality /Operator: CA
 - 1. (GI) Usability /Operator: C– /Weight: 0,2
 - 1.1. ...
 - 2. (GI) Functional Suitability /Operator: C– /Weight: 0,2
 - 2.1. ...
 - 3. (GI) Information Quality /Operator: C– /Weight: 0,3
 - 3.1. (GI) Suitability /Operator: C– /Weight: 0,7
 - 3.1.1. (GI) Coverage /Operator: C – – /Weight: 0,45
 - 3.1.1.1. (EI) Foreign language support /Weight: 0,25
 - 3.1.1.2. (EI) Line item information completeness /Weight: 0,2
 - 3.1.1.3. (EI) Product description completeness /Weight: 0,25
 - 3.1.1.4. (EI) Shipping and handling information completeness /Weight: 0,15
 - 3.1.1.5. (EI) Return policy information completeness /Weight: 0,15
 - 3.1.2. (GI) Consistency /Operator: C – – /Weight: 0,35
 - 3.1.2.1. (EI) Continue-buying feedback /Weight: 0,33
 - 3.1.2.2. (EI) Proceed-to-check-out feedback /Weight: 0,33
 - 3.1.2.3. (EI) User current status feedback /Weight: 0,33
 - 3.1.3. (GI) Value-added /Operator: DA /Weight: 0,2
 - 3.1.3.1. (EI) Recently viewed items feedback /Weight: 0,35
 - 3.1.3.2. (EI) Related items feedback /Weight: 0,35
 - 3.1.3.3. (EI) Other customers related choices feedback /Weight: 0,3
 - 3.2. (GI) Legal Compliance /Operator: A /Weight: 0,3
 - 3.2.1. (EI) Privacy policy /Weight: 0,8
 - 3.2.2. (EI) Copyright information /Weight: 0,2
 - 4. (GI) Security /Operator: CA /Weight: 0,2
 - 4.1. ...
 - 5. (GI) Reliability /Operator: C– /Weight: 0,1
 - 5.1. ...

Figura 6.3: Vista parcial del árbol de requerimientos de *Calidad Externa* mostrando el diseño de indicadores globales y elementales para el modelo LSP (sólo para el concepto calculable *Information Quality*).

El mismo proceso de diseño mediante la asignación de operadores y pesos se realiza para el resto de los indicadores globales y elementales que interpretan el modelo de concepto de *External Quality*. En la Figura 6.3 se muestran también los operadores y pesos asignados para los subconceptos y atributos del concepto *Information Quality*. Allí puede observarse que al indicador global del concepto *Suitability* se asigna un operador de simultaneidad no obligatorio *C–* con un peso de 0,7, mientras que al indicador global del concepto *Legal Compliance* se asigna el operador de neutralidad *A* con un peso de 0,3. Por otro lado, para el indicador global del concepto *Value-added* (que compone el concepto *Suitability*) se utiliza un operador de reemplazabilidad, de *cuasidisyunción* media, *DA* para reflejar el requerimiento alternativo de los atributos que lo componen.

6.4.4. Selección de los criterios de decisión

Una vez asignados los modelos elementales y globales a todos los indicadores que evalúan el modelo de concepto (conceptos calculables y atributos) definido como parte de la necesidad de información (Figura 6.2), se deben asociar *criterios de decisión* (*DecisionCriteria* –como se indica en el módulo *evaluation* presentado en la Figura 4.7) que establecen los diferentes niveles de aceptabilidad utilizados para interpretar los resultados de la evaluación y tomar las decisiones acorde a los mismos. Se recuerda, como en los casos anteriores, que los criterios de decisión son *entidades contextuales*, por lo que es posible utilizar el mecanismo de recomendación sensible al contexto para elegir el más apropiado al contexto real del proyecto.

En el catálogo de M&E se encuentran dos conjuntos de criterios de decisión. Uno de ellos corresponde a una evaluación asociada a una certificación a la cual son sometidos los proyectos de la organización anualmente por parte del grupo de empresas al que pertenece la misma. El grupo ha definido ciertos estándares para la evaluación de los proyectos de las empresas que forman parte del mismo para homogeneizar las evaluaciones y facilitar el análisis, y la posterior toma de decisiones, respecto del avance de cada una. Este criterio de decisión consta de tres niveles de aceptabilidad o certificación (tal como se muestra en la Tabla 6.26a).

Según se describe en su contexto de aplicación (representado en la Tabla 6.26b), este criterio es aplicable cuando la evaluación es destinada a comprobar el cumplimiento de estándares definidos de forma externa.

Tabla 6.26: Criterio de decisión *External certification criteria*.

(a) Especificación del criterio de decisión “*External certification criteria*”

Decision Criterion	Name	Certifies	Upper Threshold	100
	Description	Requirements established in the certification standard are fully satisfied.	Lower Threshold	85
Decision Criterion	Name	Certifies partially	Upper Threshold	85
	Description	Minimum requirements in the certification standard are satisfied. An improvement plan should be outlined and implemented in the medium-term.	Lower Threshold	60
Decision Criterion	Name	Does not certify	Upper Threshold	60
	Description	Minimum requirements in the certification standard are not satisfied. An improvement plan should be outlined and implemented in the short-term.	Lower Threshold	0

(b) Contexto de aplicación del criterio de decisión “*External certification criteria*”

CONTEXT PROPERTY			
Name	Evaluation Target	Weight	1
Relevance	This decision criteria is meant to be used in external audit certifications.	Multiplicity of Values	1
		Measures (value)	external certification
Exact Similarity Criterion		Logic Aggregator	OR
		Value Match	external certification

El otro conjunto de criterios de decisión (representados en la Tabla 6.27a) corresponde a criterios utilizados en evaluaciones internas llevadas a cabo por la misma organización como guía para mantener un nivel de calidad más estricto y detallado al utilizado en las certificaciones externas que lleva a cabo el grupo al que pertenece la organización; de esta forma la organización busca asegurarse la satisfacción de estas últimas. El contexto de aplicación asociado a estos criterios de decisión (representado en la Tabla 6.27b) refleja este aspecto, incluyendo la propiedad *Evaluation Target* con el valor “*internal improvement evaluation*”.

Observando el valor de la propiedad *Evaluation Target* en el contexto real del proyecto de M&E (Tabla 6.3) se puede ver claramente que el segundo conjunto de criterios de decisión es el seleccionado para interpretar los resultados de la evaluación en este proyecto.

Tabla 6.27: Criterio de decisión *Internal evaluation criteria*

(a) Especificación del criterio de decisión “*Internal evaluation criteria*”

Decision Criterion	Name Description	Satisfies completely	Upper Threshold	100
		Requirements are fully satisfied.	Lower Threshold	90
Decision Criterion	Name Description	Satisfies with improvements	Upper Threshold	90
		Requirements can be satisfied if improvements are applied. An improvement plan should be outlined and implemented in the medium-term.	Lower Threshold	80
Decision Criterion	Name Description	Satisfies marginally	Upper Threshold	80
		Requirements are marginally satisfied. An improvement plan should be outlined and implemented in the short-term.	Lower Threshold	60
Decision Criterion	Name Description	Does not satisfies	Upper Threshold	60
		Minimum requirements are not satisfied. An improvement plan should be outlined and implemented urgently.	Lower Threshold	0

(b) Contexto de aplicación del criterio de decisión “*Internal evaluation criteria*”

CONTEXT PROPERTY			
Name	Evaluation Target	Weight	1
Relevance	This decision criteria is meant to be used in internal evaluations.	Multiplicity of Values	1
		Measures (value)	internal evaluation
Exact Similarity Criterion		Logic Aggregator	OR
		Value Match	internal evaluation

6.4.5. Cálculo de indicadores elementales y globales

Una vez que se han diseñado todos los indicadores que evaluarán los componentes del modelo de concepto de la Figura 6.2, se lleva a cabo, en primer lugar, el cálculo correspondiente a los modelos elementales seleccionados en cada caso, y en segundo lugar del modelo global que agrupa los resultados de dichos cálculos de forma iterativa hasta el concepto *External Quality*. Cada uno de los componentes del modelo de concepto recibe un valor de preferencia que es interpretado utilizando los criterios de decisión seleccionados. El resultado de este proceso para el modelo de concepto de *External Quality* se muestra en la Figura 6.4 donde se muestran todos los conceptos calculables y atributos que componen el modelo que representa los requerimientos no funcionales enunciados, cada uno de ellos acompañado por el valor de preferencia resultante del cálculo del indicador correspondiente y la interpretación del valor según los criterios de decisión utilizados.

6.5. Análisis de los Resultados

Los resultados de la evaluación obtenidos en el punto anterior son analizados por evaluadores y gerentes de la organización para tomar acciones orientadas a la mejora de los aspectos débiles arrojados por la evaluación del modelo de concepto en sus diferentes niveles, a los cuales se corresponden las diferentes jerarquías de decisión de la organización.

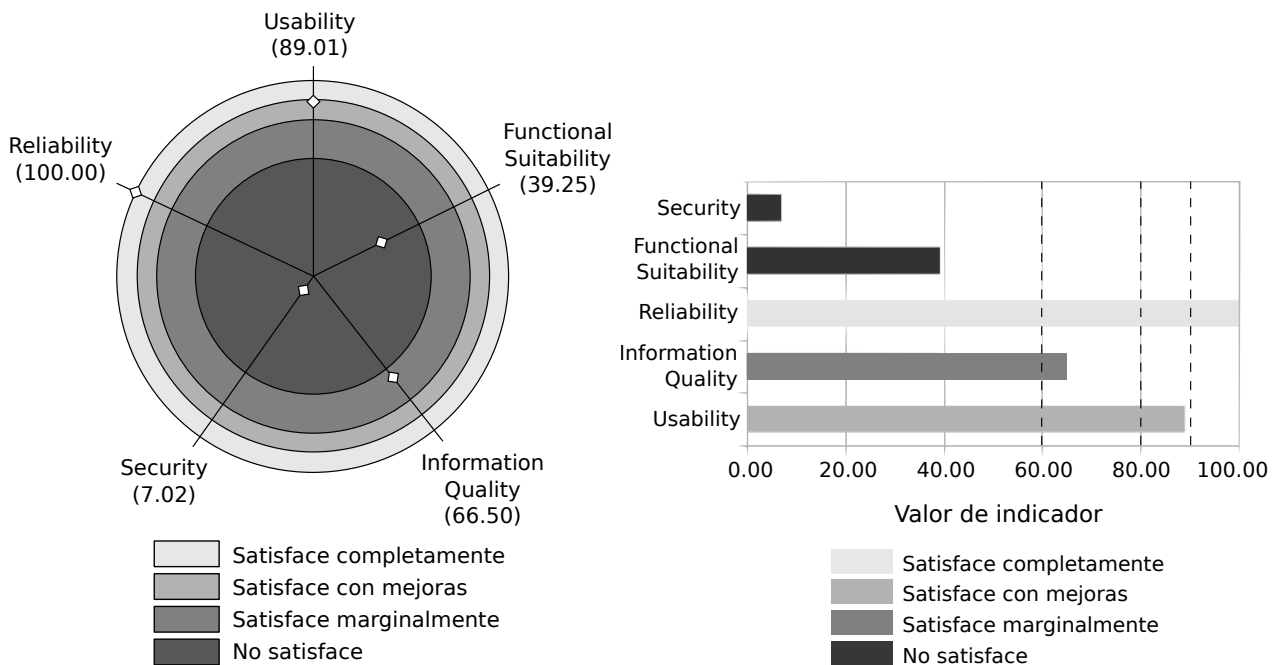
El valor de indicador global, obtenido para el concepto *External Quality*, es un valor meramente indicativo de la evaluación como un todo, y le interesa a la alta gerencia de la organización para tomar decisiones sobre la prioridad y atención que se asigne al proyecto entre el resto de los proyectos que mantiene la misma. En este caso, vemos que el valor de indicador resultante para *External Quality* es relativamente bajo considerando que el nivel de aceptabilidad “No satisface” (definido en los criterios decisión) abarca entre 0 y 60 %, por lo que el proyecto de la aplicación web Cuspide.com se encontrará entre uno de los más críticos a ser mejorados de forma urgente (según lo indica la descripción del nivel de aceptabilidad mencionado).

La gerencia del proyecto se enfoca en analizar los resultados de la evaluación en el nivel intermedio

1 Usability	89.01	Satisface marginalmente
1.1 Understandability	100.00	Satisface completamente
1.1.1 Shopping cart control ease to be recognized	100.00	Satisface completamente
1.1.2 Add-to-cart control ease to be recognized	100.00	Satisface completamente
1.2 Learnability	100.00	Satisface completamente
1.2.1 How-to-buy help availability	100.00	Satisface completamente
1.2.2 Readability of text	100.00	Satisface completamente
1.3 Operability	100.00	Satisface completamente
1.3.1 Shopping cart control permanence	100.00	Satisface completamente
1.3.2 Add-to-cart control permanence	100.00	Satisface completamente
1.3.3 Steady behavior of the shopping cart control	100.00	Satisface completamente
1.3.4 Steady behavior of the Add-to-cart control	100.00	Satisface completamente
1.4 Accesibility	28.10	No satisface
1.4.1 Images alternative text availability	50.00	No satisface
1.4.2 Support for text-only version	0.00	No satisface
2 Functional suitability	39.25	No satisface
2.1 Functional completeness	11.36	No satisface
2.1.1 Capability to add items from anywhere	50.00	No satisface
2.1.2 Capability to delete items	66.00	Satisface con mejoras
2.1.3 Capability to modify an item quantity	100.00	Satisface completamente
2.1.4 Capability to show totals by performed changes	50.00	No satisface
2.1.5 Capability to save items for later/move to cart	0.00	No satisface
2.2 Functional correctness	100.00	Satisface completamente
2.2.1 Precision to recalculate after adding an item	100.00	Satisface completamente
2.2.2 Precision to recalculate after deleting items	100.00	Satisface completamente
2.2.3 Precision to recalculate after modifying an item quantity	100.00	Satisface completamente
3 Information Quality	66.50	Satisface con mejoras
3.1 Suitability	45.96	No satisface
3.1.1 Coverage	56.25	No satisface
3.1.1.1 Foreign language support	0.00	No satisface
3.1.1.2 Line item information completeness	50.00	No satisface
3.1.1.3 Product description completeness	100.00	Satisface completamente
3.1.1.4 Shipping and handling information completeness	100.00	Satisface completamente
3.1.1.5 Return policy information completeness	100.00	Satisface completamente
3.1.2 Consistency	36.06	No satisface
3.1.2.1 Continue-buying feedback	100.00	Satisface completamente
3.1.2.2 Proceed-to-check-out feedback	50.00	No satisface
3.1.2.3 User current status feedback	0.00	No satisface
3.1.3 Value-added	39.06	No satisface
3.1.3.1 Recently viewed items feedback	0.00	No satisface
3.1.3.2 Related items feedback	50.00	No satisface
3.1.3.3 Other customers related choices feedback	0.00	No satisface
3.2 Legal Compliance	100.00	Satisface completamente
3.2.1 Privacy policy	100.00	Satisface completamente
3.2.2 Copyright information	100.00	Satisface completamente
4 Security	7.02	No satisface
4.1 Confidentiality	100.00	Satisface completamente
4.1.1 Customer registration	100.00	Satisface completamente
4.1.2 Secured check-out	100.00	Satisface completamente
4.2 Authenticity	0.00	No satisface
4.2.1 Digital certification of identity	0.00	No satisface
5 Reliability	100.00	Satisface completamente
5.1 Availability	100.00	Satisface completamente
5.1.1 Invalid links	100.00	Satisface completamente
5.1.2 Check-out service availability	100.00	Satisface completamente
5.2 Maturity	100.00	Satisface completamente
5.2.1 Unexpected results dependent of browsers	100.00	Satisface completamente
5.2.2 Unexpected results independent of browsers	100.00	Satisface completamente

Figura 6.4: Evaluación del modelo de concepto de *Calidad Externa*.

del modelo de concepto, distinguiendo entre los diferentes conceptos calculables que componen External Quality –*Usability*, *Functional Suitability*, *Information Quality*, *Security* y *Reliability*. En este aspecto, los gerentes y evaluadores pueden valerse de gráficos que representen los valores de tales conceptos de manera que permitan ser contemplados de forma conjunta y comparados entre sí para determinar posibles planes de acción. Estos gráficos pueden ser generados rápidamente de forma automática utilizando herramientas apropiadas. En la Figura 6.5 se presentan dos tipos de gráficos que pueden ser utilizados para tal propósito. En el gráfico de la Figura 6.5a se presentan los valores de indicador para los conceptos mencionados en un diagrama de red circular, distinguiendo en zonas concéntricas los diferentes criterios de decisión o niveles de aceptabilidad, en los cuales hacia el centro se encuentran los valores más bajos y hacia afuera los más altos. De forma similar, en el gráfico de la Figura 6.5b se observan los mismos valores pero representados en un gráfico de barras horizontales (que representa la evaluación resultante para cada concepto), también distinguiendo los diferentes niveles de aceptabilidad (mediante líneas punteadas) pudiendo distinguir fácilmente la magnitud relativa entre los valores de indicador de uno y otro concepto.



(a) Resultados de la evaluación en un gráfico de red en zonas. (b) Resultados de la evaluación en un gráfico de barras.

Figura 6.5: Diferentes representaciones de los resultados de alto nivel de la evaluación de *External Quality* de Cuspide.com.

Del análisis de estos gráficos puede observarse fácilmente que los conceptos más críticos a ser mejorados son *Security*, *Functional Suitability* e *Information Quality* –aquellos que se encuentran en los niveles “*No satisface*” y “*Satisface marginalmente*”–, para los cuales se requiere realizar un análisis más detallado del porqué del bajo valor de evaluación obtenido. En el caso de *Information Quality* puede observarse que (viendo la Figura 6.4), de ambos conceptos que la conforman, sólo *Legal Compliance* fue satisfecho en un 100 % mientras *Suitability* sólo en un 45,96 %. Por lo tanto resta analizar los resultados de la evaluación para los componentes de éste último concepto. En este punto también se puede recurrir a gráficos similares a los presentados antes para enfocar rápidamente los esfuerzos en los componentes más críticos del modelo. Como ejemplo, en la Figura 6.6 se representan los resultados de la evaluación obtenidos para los conceptos que componen *Suitability*. En este gráfico puede observarse

que el concepto *Consistency* ha recibido la evaluación más baja. Consultando los resultados de la evaluación (Figura 6.4) puede verse que el atributo *User current status feedback* ha sido evaluado en 0% (la aplicación no provee información visible del estado de la compra del usuario cliente mientras navega por el catálogo de productos del sitio). Luego, en el concepto *Value-added* se observan dos atributos que reciben una evaluación también de 0%, *Recently viewed items feedback* y *Other customers related choices feedback*. Finalmente, para el concepto *Coverage* el único atributo evaluado en 0% es *Foreign language support* (el sitio no puede ser visto en lenguajes extranjeros).

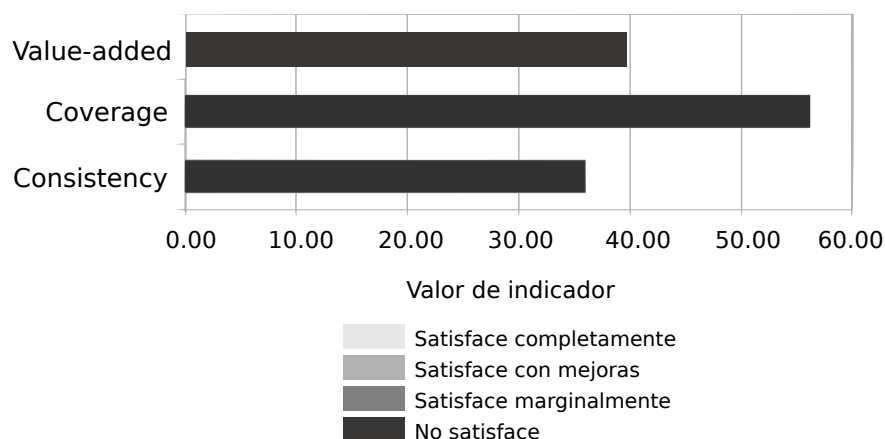


Figura 6.6: Representación de resultados parciales de la evaluación de *External Quality* –para los conceptos que componen *Suitability* (en *Information Quality*).

Este análisis se repite para todos los conceptos del modelo de concepto tras el cual se efectúa un plan de mejoras que priorice los componentes del modelo que hayan recibido la menor evaluación, considerando, como en toda empresa, un balance entre costos y beneficios adecuado a la situación actual. En éste sentido, la documentación generada a partir de la descripción del contexto real del proyecto puede ser utilizada en todo momento para fundamentar las conclusiones obtenidas y justificar las decisiones que se lleven a cabo.

Más allá de los resultados y conclusiones obtenidos en este análisis, los diseños de medición y evaluación realizados en el proyecto pueden ser reutilizados en el futuro para repetir la misma evaluación a la entidad y realizar comparaciones con los resultados previos para determinar si las acciones implementadas a partir de los resultados de la primer instancia de evaluación han producido mejoras efectivas en los requerimientos no funcionales especificados originalmente. También es posible repetir la medición y evaluación del proyecto original para nuevas entidades, en principio, de la misma categoría de entidad a la evaluada originalmente, para efectuar análisis comparativos entre diferentes productos. Los resultados de esta nueva instancia de evaluación serán compatibles con los primeros (es decir que pueden ser coherentemente comparados) en la medida en que el contexto relevante de la nueva instancia del proyecto no varíe respecto del primero, para lo cual se dispone de la descripción estructurada del contexto real de cada proyecto y del mecanismo que permite determinar la similitud entre dos descripciones de contexto. Si tal coherencia entre resultados de diferentes proyectos no puede ser comprobada o asegurada, las conclusiones a las que se arriben a partir de una comparación pueden ser erróneas y llevar, en consecuencia, a tomar decisiones inapropiadas.

6.6. Conclusiones

A lo largo de este capítulo hemos mostrado, mediante una prueba de concepto, cómo es posible utilizar el marco C-INCAMI para llevar a cabo la especificación de los metadatos y datos necesarios para definir e implementar las actividades de medición y evaluación de calidad de una entidad de interés para una organización de software, tal como la aplicación web *Cuspide.com*.

Así, utilizando el marco C-INCAMI ha sido posible especificar de forma clara y estructurada la necesidad de información que motiva y guía la realización de las tareas posteriores de medición y evaluación. Asociado a dicha información, el marco permitió describir, también de forma estructurada, el contexto relevante en el cual se llevan a cabo las tareas mencionadas. Esta información establece un marco de referencia que permite guiar el diseño e implementación de la medición y evaluación de forma consistente y coherente para satisfacer la necesidad de información de forma adecuada. También se pudo ver cómo el marco C-INCAMI permitió realizar una especificación explícita y cuantificable del concepto foco de interés de la evaluación, mediante el uso de modelos de conceptos –en la forma de un árbol de requerimientos–, estableciendo de forma precisa el “qué” de la evaluación. Luego, se mostró que el marco permite especificar de forma explícita el “cómo” se lleva a cabo la cuantificación de los atributos, que permite satisfacer la necesidad de información de bajo nivel, mediante el uso de métricas claramente estructuradas. De la misma forma fue posible registrar los resultados de dicha medición, especificando los metadatos apropiados. Luego, el marco permitió definir el “cómo” respecto de la satisfacción de la necesidad de alto nivel, utilizando indicadores (y modelos asociados) que describen explícitamente cómo convertir los valores medidos de los atributos en valores de preferencia o desempeño que, una vez obtenidos, responden a la “pregunta” formulada inicialmente por la necesidad de información. Todas estas actividades pueden ser llevadas a cabo en un marco de coherencia y consistencia provisto por la especificación explícita del contexto real del proyecto de M&E, de los requerimientos no funcionales, métricas e indicadores, reduciendo en gran medida los errores que pudieran producirse por la mal interpretación de las tareas a realizar o de los resultados obtenidos.

También se ha mostrado que el uso del marco C-INCAMI permite disponer de catálogos organizacionales estructurados que contengan especificaciones de los elementos de información reusables durante las tareas de diseño y especificación (atributos, métricas, modelos), promoviendo la consistencia entre los diferentes proyectos de M&E de la organización. Esta consistencia permite llevar a cabo, de forma coherente, comparaciones entre los diferentes proyectos de la organización y entre distintos momentos históricos de un mismo proyecto, permitiendo llegar a conclusiones más validas, que se resumen en la toma de decisiones más robusta ya que se basan en resultados contextualizados, especificados y documentados de forma clara y explícita.



Soporte Tecnológico a C-INCAMI



Como se indicó en el Capítulo 1, para que una organización pueda asegurar la calidad de sus procesos y productos de software y web debe llevar a cabo programas de medición y evaluación orientados a satisfacer sus metas y objetivos. Para lograr esto, la organización necesita establecer un conjunto de actividades y procedimientos para especificar, reunir, registrar y utilizar datos que describan de forma clara, consistente y coherente el estado y situación de las entidades críticas de su operatoria. En este sentido, la especificación de metadatos y datos de necesidades de información, contextos, requerimientos no funcionales, métricas e indicadores –entre otros relacionados- proveen los cimientos básicos para contar con información robusta para llevar adelante proyectos mas repetibles y controlables. En esta empresa, el uso de herramientas que den soporte tecnológico a tales actividades es esencial a la luz de la diversidad y complejidad de aspectos que influyen en la calidad de los productos y servicios ofrecidos por una organización, más aun considerando los beneficios que proveen las tecnologías de información en la actualidad para registrar y transmitir el conocimiento.

En este capítulo se describen los aspectos de diseño e implementación de una herramienta, llamada C-INCAMI^{PM} (*Project Manager*) que da soporte tecnológico al marco C-INCAMI propuesto en esta tesis: la aplicación permite gestionar la ejecución de proyectos de M&E de calidad siguiendo las actividades definidas por el proceso de M&E citado en la Sección 3.2.2 y los lineamientos de la metodología de evaluación WebQEM descrita en la Sección 3.2.3, permitiendo especificar los metadatos y datos de cada etapa tal como han sido definidos en el marco C-INCAMI (como se describió a lo largo de los Capítulos 4, 5 y 6). La arquitectura propuesta para la herramienta permite integrarla a los sistemas informáticos de la organización que mantienen datos asociados al dominio de aplicación de la misma. De esta forma, se facilita la incorporación de la herramienta y, más importante, de un base conceptual robusta para establecer una capacidad de medición y evaluación en la organización, instrumento esencial en la gestión de proyectos de software orientada a la mejora. En conclusión, la herramienta pretende, mediante la especificación clara y precisa de los metadatos y datos asociados a la M&E de requerimientos no funcionales de entidades de software y web, asegurar que los valores medidos y las evaluaciones obtenidas sean repetibles y comparables entre los proyectos de la organización de forma consistente, resultando en un instrumento robusto para la toma decisiones hacia la mejora de los productos y servicios de software y web de la organización.

En el resto del capítulo se establece el alcance de la funcionalidad a la que da soporte la arquitectura propuesta y luego se describe cada una de las capas y módulos de la arquitectura, cubriendo aspectos tecnológicos, tal como fueron implementados en un prototipo que materializa dicha arquitectura.

7.1. Alcance

La herramienta C-INCAMI^{PM} da soporte a las actividades de especificación de requerimientos, diseño e implementación de la medición y diseño e implementación de la evaluación usando las definiciones del marco C-INCAMI. La herramienta interactúa con un catálogo de M&E, disponible a todos los proyectos de la organización, del cual se recuperan elementos de información de M&E reusables, especificados a partir de los conceptos definidos en el marco C-INCAMI. La herramienta hace uso de repositorios de información existentes en la organización para importar definiciones y especificaciones de los elementos de información relevantes –categorías de entidad, entidades, atributos y propiedades de contexto– propios del dominio de aplicación de la misma, al catálogo de M&E. Finalmente, la herramienta diseñada permite almacenar los resultados del proceso en memorias de proyectos de M&E con fines de análisis comparativos e históricos.

7.2. Arquitectura

La arquitectura de la herramienta para el soporte tecnológico del marco C-INCAMI debía satisfacer un conjunto de requerimientos establecidos implícitamente por los enfoques de M&E y sensible al contexto adoptados, tal como se describieron en los capítulos 3, 4 y 5. Concretamente, debía satisfacer:

- la transferencia efectiva de los diseños conceptuales realizados con el paradigma *orientado a objetos*;
- la posibilidad de acceder a la herramienta desde cualquier nodo/terminal disponible para cada proyecto;
- la consistencia entre los diseños y resultados de M&E de todos los proyectos de la organización;
- el mecanismo de integración de las especificaciones realizadas con el marco C-INCAMI al espacio de información del dominio de aplicación de la organización (descrito en la Sección 4.6);
- la fácil instalación y despliegue de los componentes de la herramienta en los sistemas de la organización;
- la maximización en el reuso de la infraestructura de procesamiento y comunicación existente en la organización.

Para satisfacer estos requerimientos se recurre a una arquitectura cliente/servidor distribuida, en la que los componentes se distribuyen en un número de capas alojadas en diferentes nodos de una red (local o global) con el objetivo de lograr una clara separación de los distintos aspectos funcionales del sistema, ofreciendo a la vez un nivel de modularidad que permite el reuso de los distintos componentes. El modelo de cliente utilizado es el de cliente liviano. En este modelo, el cliente es responsable únicamente de mostrar la interfaz gráfica generada por –y enviar solicitudes a– un servidor de aplicación remoto del cual utiliza los recursos de procesamiento de datos necesarios. A este modelo de cliente pertenecen las aplicaciones web que utilizan tecnologías tales como HTML, JavaScript/AJAX , ASP, JSP, PHP, entre otros. El diseño de la herramienta como una aplicación web accediendo a diferentes servidores de aplicación remotos permite aprovechar la arquitectura de comunicación y procesamiento provista por internet y las tecnologías web, particularmente la web semántica –de la cual depende el enfoque diseñado–, y además para facilitar la instalación y despliegue de los componentes en los sistemas de la organización.

La arquitectura se divide en cuatro capas: presentación, lógica de negocios, lógica de acceso a datos y persistencia –tal como se muestra en la Figura 7.1.

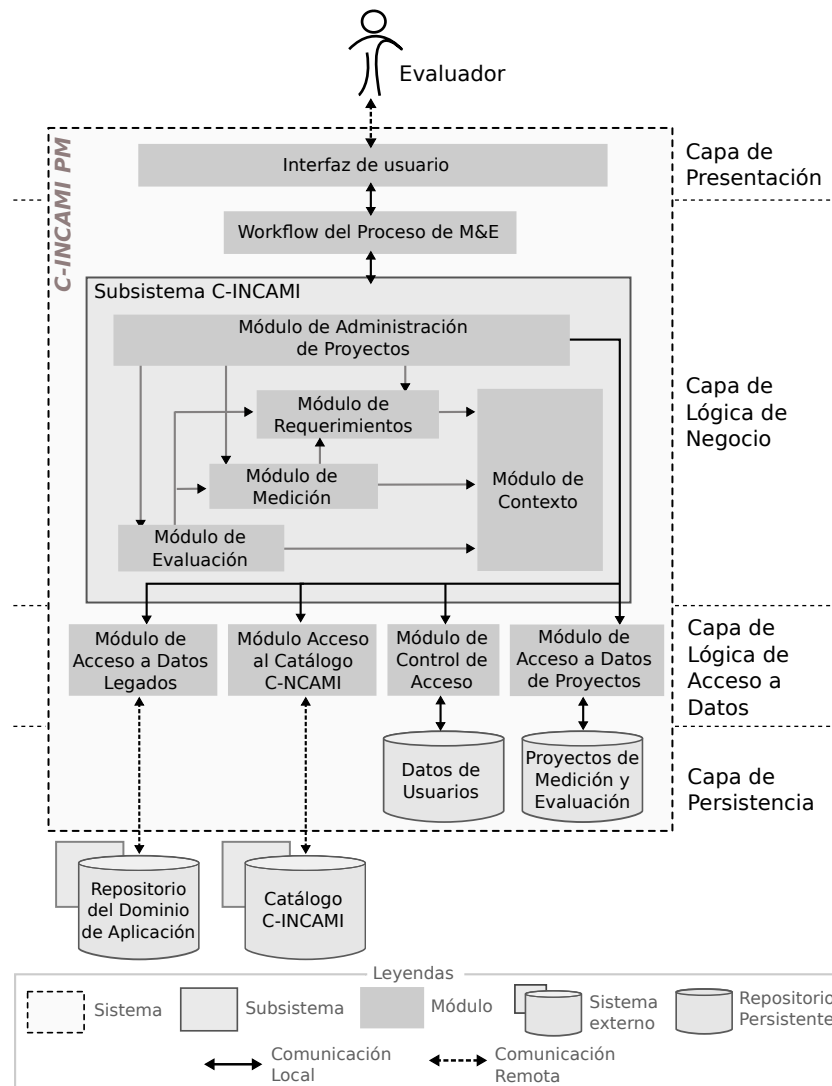


Figura 7.1: Arquitectura de C-INCAMI Project Manager

La **capa de presentación** consiste de una interfaz web compuesta por un conjunto de páginas dinámicas, que permite a un usuario evaluador acceder a la funcionalidad para llevar a cabo las tareas correspondientes a cada fase del proceso de M&E. Esta capa actúa sobre (y es actualizada por) la capa de lógica de negocios.

En la **capa de lógica de negocio** se encuentran los componentes que implementan la funcionalidad asociada a las actividades de M&E. Esta capa incluye los módulos correspondientes al marco C-INCAMI, responsables de implementar la estructura y el comportamiento inherente a cada uno de los conceptos y relaciones definidos en el mismo. Además incluye un módulo que implementa la lógica que guía al usuario evaluador en el proceso de completar las fases de definición y especificación de requerimientos, y de diseño e implementación de la medición y la evaluación (el *workflow* del proceso) descrito brevemente en la Sección 3.2.2. Los componentes de esta capa hacen uso de la capa de *lógica de acceso a datos* para ofrecer servicios de persistencia.

En la **capa de lógica de acceso a datos** se encuentran los componentes que ofrecen el servicio de escribir en (guardar), y leer desde (recuperar), un medio de almacenamiento persistente apropiado, los datos utilizados por la herramienta. El objetivo de esta capa es procesar los datos manipulados por la herramienta apropiadamente para su almacenamiento/recuperación y ocultar la complejidad inherente a la manipulación de tecnologías de almacenamiento persistente de información (de la capa subyacente), necesaria para recuperar y almacenar los objetos de datos manipulados por la herramienta.

La **capa de persistencia** incluye los componentes encargados de almacenar de forma persistente los datos utilizados y generados por la herramienta. Éstos corresponden a cuatro aspectos del sistema (de izquierda a derecha en el diagrama de la Figura 7.1):

- el repositorio de la información del dominio de negocios de la organización, que contiene los metadatos (definiciones de los términos y relaciones entre ellos) y datos asociados a su operatoria. Este repositorio es propio de la organización y por lo tanto se considera un sistema externo;
- el catálogo de M&E que contiene los metadatos y datos correspondientes a los conceptos y relaciones definidos en el marco C-INCAMI (conceptos, atributos, métricas, etc. –descriptos en el Capítulo 4) disponible a todos los proyectos de M&E de la organización. Este catálogo utiliza tecnología de web semántica para procesar consultas y almacenar los datos y metadatos del marco C-INCAMI;
- los datos de los usuarios evaluadores y los proyectos dirigidos por ellos, utilizados para el control de acceso a la herramienta. Este repositorio utiliza tecnología de almacenamiento relacional;
- las datos y metadatos completos de los proyectos de M&E llevados a cabo, que incluyen las especificaciones de requerimientos no funcionales, diseño y resultados de medición y evaluación. Estos datos y metadatos mantienen las referencias correspondientes a los datos y definiciones almacenadas en el catálogo de M&E. Este repositorio almacena la información mencionada utilizando un formato estructurado abierto e intercambiable –XML (*eXtensible Markup Language*) [W3C-XML, 2012].

A continuación se describe con más detalle cada uno de los componentes de la arquitectura diseñada para la herramienta C-INCAMI^{PM} describiendo, en cada caso, las funciones que llevan a cabo, las relaciones con otros componentes y aspectos tecnológicos de su implementación.

7.2.1. C-INCAMI Project Manager

La herramienta C-INCAMI^{PM} fue desarrollada como una aplicación web utilizando *Barracuda MVC*, un marco de presentación de código abierto (*open source*), basado en la tecnología Servlet [Jendrock *et al.*, 2012] de JavaTM, diseñado para facilitar la construcción de aplicaciones web. Este marco se caracteriza por: (i) un mecanismo de plantillas y un modelo de componentes de interfaz gráfica basados en *DOM* [W3C-DOM, 1998] y *XMLC* [XMLC, 2004] que implementan un patrón *MVC* (model-view-controller) para aplicaciones web, similar al de *Swing* –paquete de interfaz gráfica de JavaTM, (ii) un modelo de eventos basado en *Model 2* [Seshadri, 1999], que soporta la programación dirigida por eventos en el paradigma web (todos ellos descriptos en el Cuadro 7.1). De esta forma, es posible agregar oyentes a los controles de manera que, ante una acción en el cliente, se generan eventos reales en el servidor notificando a los manejadores de eventos asociados que procesan el evento. Este modelo resulta en un débil acoplamiento con mayor cohesión,

Cuadro 7.1. Tecnologías para el desarrollo de aplicaciones web.

XML (*eXtensible Markup Language*) es un metalenguaje extensible de etiquetas desarrollado por el W3C, como una simplificación y adaptación del *SGML*, que permite definir la gramática de dominios específicos y describir información acerca de estos utilizando un formato de texto estructurado. El vocabulario (las reglas sintácticas) *XML* de un dominio puede definirse mediante *DTDs* o *XML Schemas*. Como tal, *XML* fue propuesto como un estándar para el intercambio de información estructurada entre diferentes plataformas, alrededor del cual se implementan otras tecnologías que la complementan permitiendo la compatibilidad entre diversos sistemas para compartir información fácilmente.

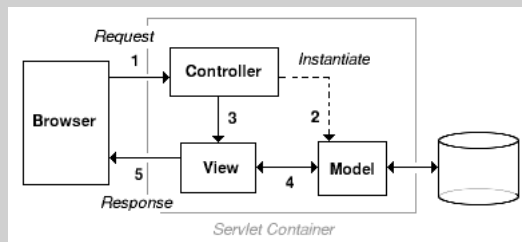
▷ **DOM** (*Document Object Model*). Modelo de objetos estándar e independiente del lenguaje para representar documentos HTML/XML; Diferentes implementaciones del modelo permiten acceder a los elementos de dichos documentos, tal como el API JavaTM DOM que provee interfaces genéricas para XML [Java API DOM-XML, 2003], como por ejemplo: *Node*, *Document*, *Element*, *Text*, así como para HTML [Java API DOM-HTML, 2003], como *HTMLDocument*, *HTMLHeadElement*, *HTMLBodyElements*, *HTMLAnchorElement*, entre otros.

controlando fácilmente el flujo de control; y (iii) un conjunto de paquetes que permiten la validación y mapeo de solicitudes HTTP de formularios a objetos de forma automática, así como el soporte de internacionalización.

Cuadro 7.1. Tecnologías para el desarrollo de aplicaciones web (cont.)

▷ **XMLC** (*XML Compiler*). Mecanismo que permite generar, a partir de un documento HTML o XML estático (sin necesidad de código embebido), una clase XMLC que contiene una representación DOM del documento. Los objetos DOM de la clase XMLC pueden luego ser instanciados y “renderizados” dinámicamente en el servidor con los datos apropiados y enviados al cliente como documentos HTML/XML (estrategia conocida como Push-MVC).

▷ **Model 2**. Arquitectura MVC para aplicaciones Web en la que las solicitudes HTTP son pasadas desde el cliente a un Servlet (Java™) controlador, quien actualiza el modelo y luego invoca al generador de vista apropiado que obtiene los datos a mostrar desde el modelo actualizado (ver figura).



Flujo de control de la arquitectura Model 2.

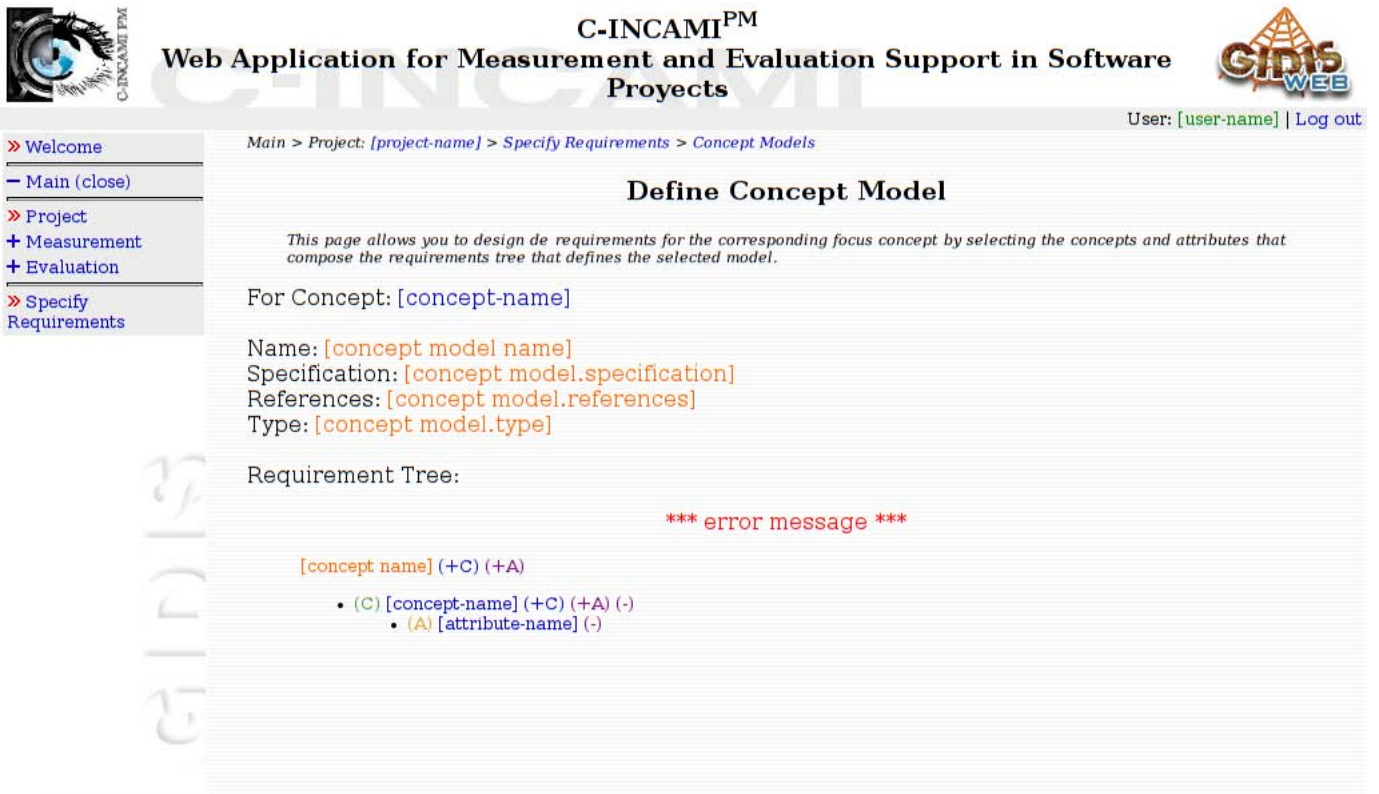
La herramienta se compone de un conjunto de módulos que implementan cada una de las funciones correspondientes a las capas de la arquitectura presentada en la Figura 7.1, más dos repositorios persistentes que almacenan información propia de la herramienta (como se describirá luego en el punto 7.2.1.4). Además accede, también de forma remota, a dos sistemas externos, cada uno con sus módulos de acceso correspondientes, independientes de la implementación de C-INCAMI^{PM}. Estos dos sistemas corresponden al *Catálogo C-INCAMI* (descrito en la Sección 7.2.2) y el repositorio de información del dominio de aplicación de la organización (descrito en la Sección 7.2.3).

La herramienta puede ser accedida de forma remota (dentro de una intranet o globalmente), a través de una interfaz web, por un usuario evaluador, previa validación para comprobar su identidad (ver *Módulo de Control de Acceso* en el punto 7.2.1.3). A continuación se describen los módulos de C-INCAMI^{PM} con más detalle.

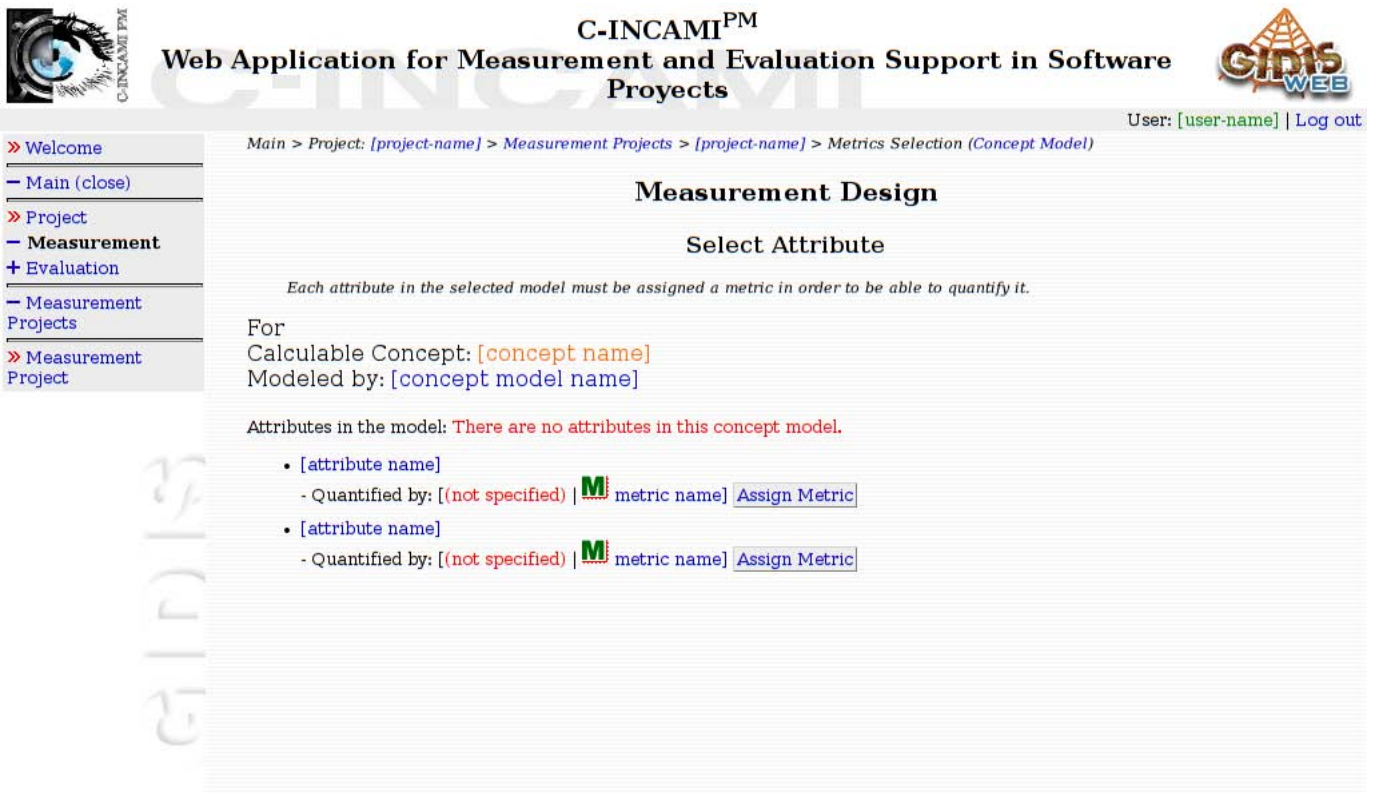
7.2.1.1. Capa de Presentación

Módulo de Interfaz de usuario En este módulo se encuentran un conjunto de componentes (clases Java™) utilizados para generar o “renderizar” dinámicamente la interfaz que será enviada al navegador web del cliente como páginas HTML, scripts e imágenes. Como se mencionó anteriormente, estos componentes consisten en clases XMLC, que representan las diferentes vistas de la herramienta utilizando un modelo DOM. Las clases XMLC son generadas utilizando como entrada un conjunto de páginas HTML estáticas, que representan las diferentes vistas de la herramienta para cada una de las tareas del proceso de M&E. Estas páginas son diseñadas y enlazadas para establecer la apariencia del sistema así como los caminos de navegación que reflejan el *workflow* del proceso de M&E (presentado en la Sección 3.2.2). En la Figura 7.2 se muestran dos de las páginas diseñadas para tal fin: la primera (Figura 7.2a) permite construir el árbol de requerimientos como parte del diseño de un modelo de concepto; la segunda (Figura 7.2b) corresponde a la tarea de asignación de métricas a los diferentes atributos de un modelo de concepto, como parte del diseño de la medición. También, en la Figura 7.3, se muestra un diagrama que ilustra los enlaces de navegación principales¹ entre todas las páginas diseñadas de la herramienta C-INCAMI^{PM}.

¹No se incluyen los enlaces que permiten retornar a actividades anteriores.



(a) Plantilla HTML utilizada en la generación del objeto XMLC que permite mostrar y diseñar un Modelo de Concepo.



(b) Plantilla HTML utilizada en la generación del objeto XMLC que permite seleccionar las métricas para cada Atributo de un Modelo de Concepto.

Figura 7.2: Páginas HTML estáticas utilizadas como plantillas para la generación de las vistas dinámicas de la herramienta C-INCAMI^{PM}.

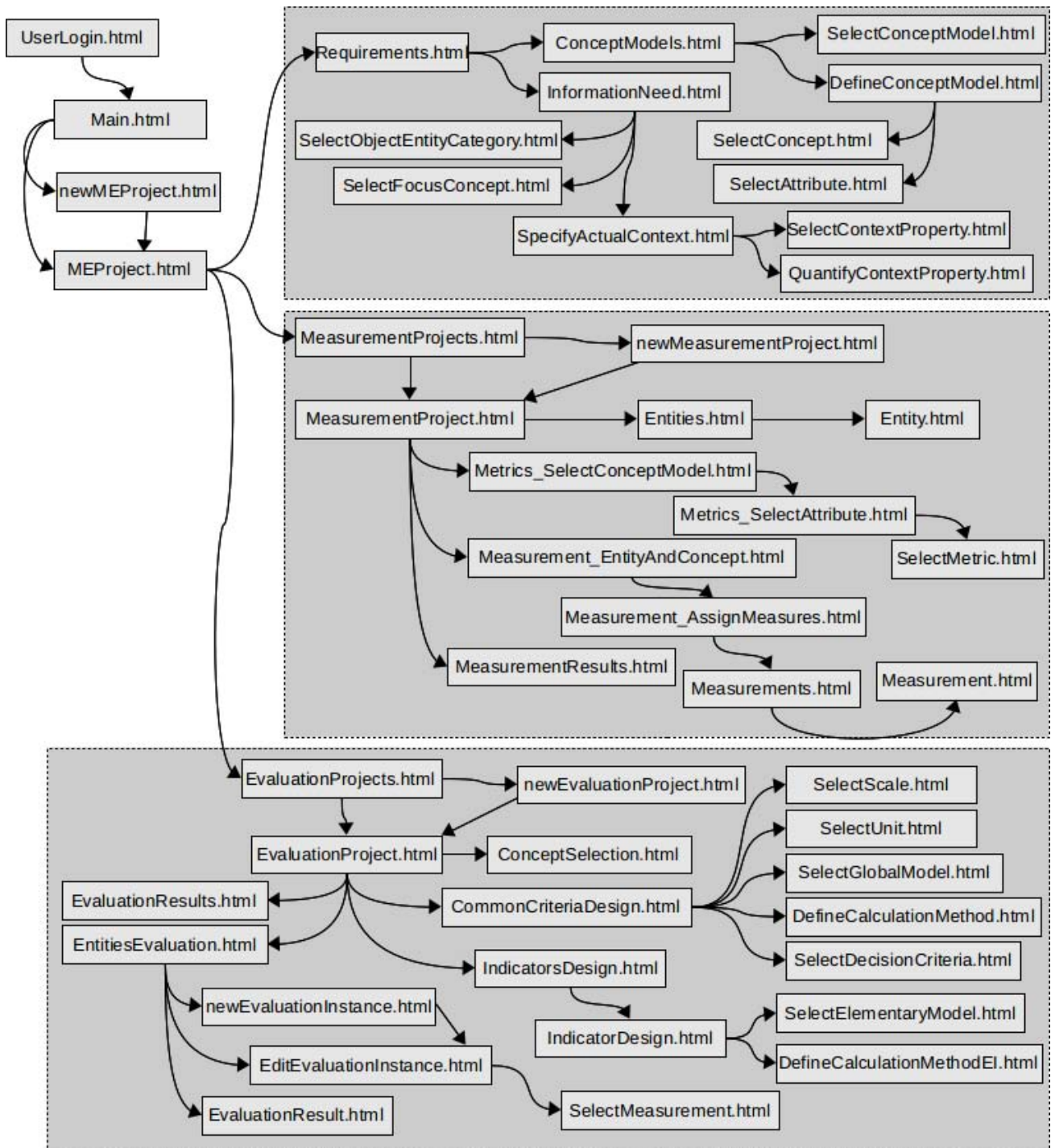


Figura 7.3: Páginas estáticas y enlaces de navegación de la aplicación web C-INCAMI^{PM}.

Para cada página <nombre-pagina>.html el compilador XMLC genera una clase Java <nombre-pagina>HTML.class. Las clases generadas a partir de estas plantillas HTML, conforman el paquete `gidisweb.c_incami.presentation.xmlc`² que constituye el componente principal del módulo de interfaz de usuario.

La estructura DOM de las clases XMLC generadas será manipulada por la lógica de negocios para generar dinámicamente la interfaz web del cliente. Los elementos de las páginas generadas dinámicamente y enviadas al cliente pueden contener información asociada a los eventos que serán disparados

²El prefijo "gidisweb" identificar al grupo de investigación en cual se llevó a cabo la investigación. El cambio de nombre c-incami a c_incami se debe a una regla de sintaxis del lenguaje JavaTM.

en el servidor al efectuar solicitudes HTTP. La generación y procesamiento de estos eventos es responsabilidad del módulo de la capa inferior (*Workflow del Proceso de M&E*).

7.2.1.2. Capa de Lógica de Negocio

La *capa de lógica de negocios* está compuesta, por un lado, por los componentes que implementan la lógica que guía al usuario a través de las diferentes tareas involucradas en las actividades de diseño e implementación de M&E (lógica de navegación) y, por otro lado, por los componentes que constituyen el corazón del sistema, proveyendo el comportamiento inherente a todos los conceptos involucrados en dichas actividades (la implementación del marco C-INCAMI propiamente dicho).

Módulo del Workflow del Proceso de M&E Este módulo implementa, a través de un conjunto de componentes, la lógica de navegación de la aplicación web que guía al usuario a través de las diferentes tareas involucradas en las actividades, siguiendo el *workflow* definido por el proceso de M&E presentado en la Sección 3.2.2. Para ello, extiende el comportamiento de los componentes del *framework* Barracuda MVC encargados de la recepción de solicitudes HTTP, procesamiento de eventos, validación de formularios, manipulación de vistas XMLC y envíos de respuestas al cliente (entre otros). Los siguientes componentes conforman este módulo (ver Figura 7.4):

`gidisweb.c_incami`: Este componente (a través de su única clase `C_INCAMIGateway`) se encarga de recibir las solicitudes HTTP del cliente web y convertirlas a eventos (objetos) específicos definidos para la aplicación que serán capturados por el motor de eventos del *framework* Barracuda MVC. Para ello, declara y registra un conjunto de manejadores para todos los eventos posibles.

`gidisweb.c_incami.presentation.events`: Este componente contiene las clases correspondientes a los eventos que pudieran ser generados desde el cliente así como aquellos que pueden ser disparados por otros eventos –tales como *GetProjects*, *ShowEntities*, *DoSelectMetric* y *DoEvaluate*, entre otros. Estas clases son generadas automáticamente por el *framework* Barracuda MVC a partir de un archivo de configuración en formato XML.

`gidisweb.c_incami.presentation`: Se conforma principalmente de un número de oyentes y manejadores de eventos, cada uno de los cuales es responsable de procesar un conjunto de eventos relacionados entre sí. También incluye un conjunto de clases que permiten acceder a los objetos de la lógica de negocios (los modelos del patrón *model-view-controller*) así como a los formularios HTML presentes en las diferentes vistas, facilitando un mecanismo de validación de parámetros. Las clases definidas en este paquete son responsables de:

- procesar los eventos capturados llevando a cabo la lógica de negocio (las actividades del proceso de M&E) asociada al evento disparado, para lo cual interactúan con los módulos del subsistema C-INCAMI, manipulando –o recuperando de los repositorios apropiados- las especificaciones de los conceptos definidos en el Capítulo 4,
- instanciar las vistas XMLC del módulo de interfaz de usuario (utilizando el API DOM de Java™) y generar las páginas correspondientes que serán enviadas al cliente en respuesta a las solicitudes realizadas por este último.

Subsistema C-INCAMI Este subsistema está constituido por los módulos que conforman la implementación del marco conceptual C-INCAMI, tal como se describieron en los capítulos 4 y 5, más un conjunto de componentes que actúan de intermediarios entre dichos módulos y el módulo de *workflow* del proceso de M&E. Cada módulo del marco C-INCAMI es implementado como un paquete

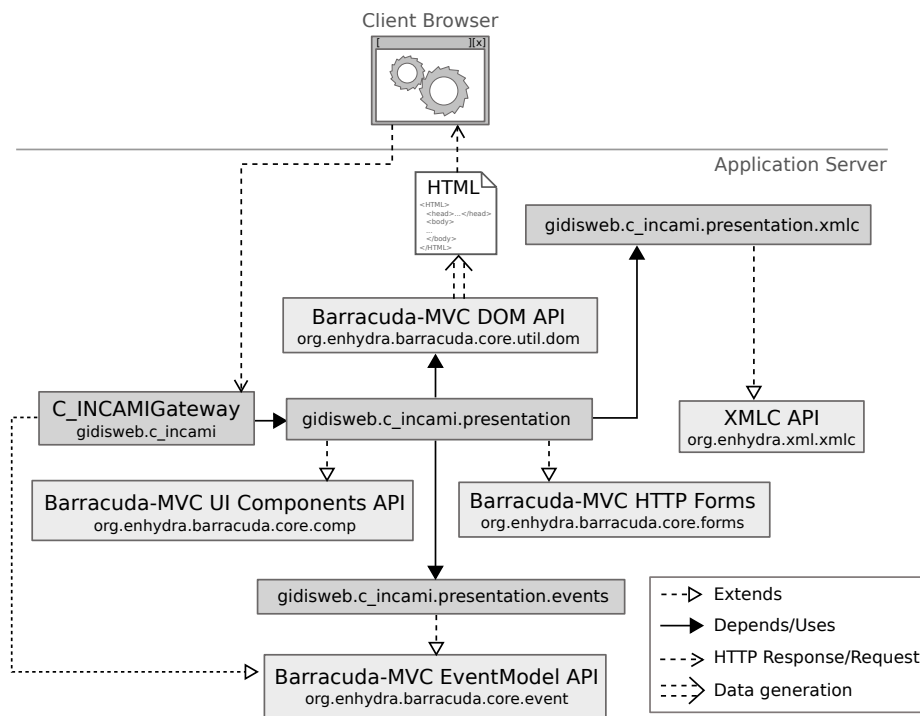


Figura 7.4: Componentes que integran el módulo de Workflow del Proceso de M&E.

Java™ conteniendo las clases (y relaciones) pertenecientes al módulo correspondiente. Cada clase provee la definición de la estructura y la funcionalidad asociada al concepto correspondiente. A excepción de los componentes que actúan como intermediarios de la capa superior, éstos paquetes fueron desarrollados de manera independiente de cualquier otro aspecto de implementación del sistema, tratando de mantener de esta forma, la modularidad buscada para cada una de las capas del mismo. Así, estos módulos pueden ser reusados por otros sistemas reemplazando, modificando o agregando nuevas capas de software superiores o adyacentes a las implementadas aquí.

Los módulos o paquetes que contiene este subsistema son, tal como se especificaron en el Capítulo 4 (recordar la Figura 4.1), `gidisweb.c_incami` (el paquete principal que incluye la gestión de proyectos de M&E), `gidisweb.c_incami.requirements`, `gidisweb.c_incami.context`, `gidisweb.c_incami.measurement` y `gidisweb.c_incami.evaluation`. Las clases intermediarias entre el módulo de workflow de proceso de M&E y los paquetes que implementan el marco C-INCAMI forman parte del paquete `gidisweb.c_incami`, y son las siguientes:

- **C_INCAMIFramework:** provee un único punto de acceso a la funcionalidad implementada en los módulos del marco C-INCAMI. Es responsable de validar solicitudes de acceso de usuarios evaluadores –utilizando el módulo de acceso a datos correspondiente–, para cada uno de los cuales construye un espacio de trabajo (*Workspace*) dentro del cual se gestionarán los proyectos de M&E.
- **Workspace:** mantiene referencia y provee acceso a un proyecto de M&E activo (*MEProject*), siendo responsable de la interacción con los diferentes repositorios subyacentes –por medio de los módulos de acceso a datos– para proveer funciones de lectura, escritura y consulta de información relevante a dicho proyecto. Mantiene además la referencia a los datos del usuario evaluador responsable de la gestión de los proyectos de M&E que sean creados y/o editados (*RegisteredUser*).
- **RegisteredUser:** mantiene los datos de un usuario evaluador validado por la aplicación para acceder a los recursos de los módulos del marco C-INCAMI.

A continuación se describe la funcionalidad implementada por cada uno de estos módulos, propia del marco C-INCAMI:

Administración de Proyectos de M&E Luego de validarse como usuario registrado, un evaluador podrá (según se refleja en el diagrama de casos de uso de la Figura 7.5) crear y gestionar proyectos de M&E (*MEProject*). Esta gestión incluye la especificación de requerimientos no funcionales y la creación y gestión de subproyectos de medición y subproyectos de evaluación. Los proyectos de M&E creados pueden ser guardados en el repositorio de proyectos de la organización para su posterior recuperación, edición (para continuar con su gestión) y análisis de resultados. La gestión de proyectos también incluye la posibilidad de eliminar proyectos guardados previamente. Cabe aclarar, como se muestra en la figura citada, que la gestión de subproyectos de medición requiere³ que la especificación de requerimientos no funcionales haya sido completada; de la misma forma, la gestión de subproyectos de evaluación requiere completar la gestión de (al menos) un subproyecto de medición. En las Figuras 7.6 y 7.7 se muestran capturas de pantalla de la herramienta C-INCAMI^{PM} de las vistas de gestión de proyectos de M&E.

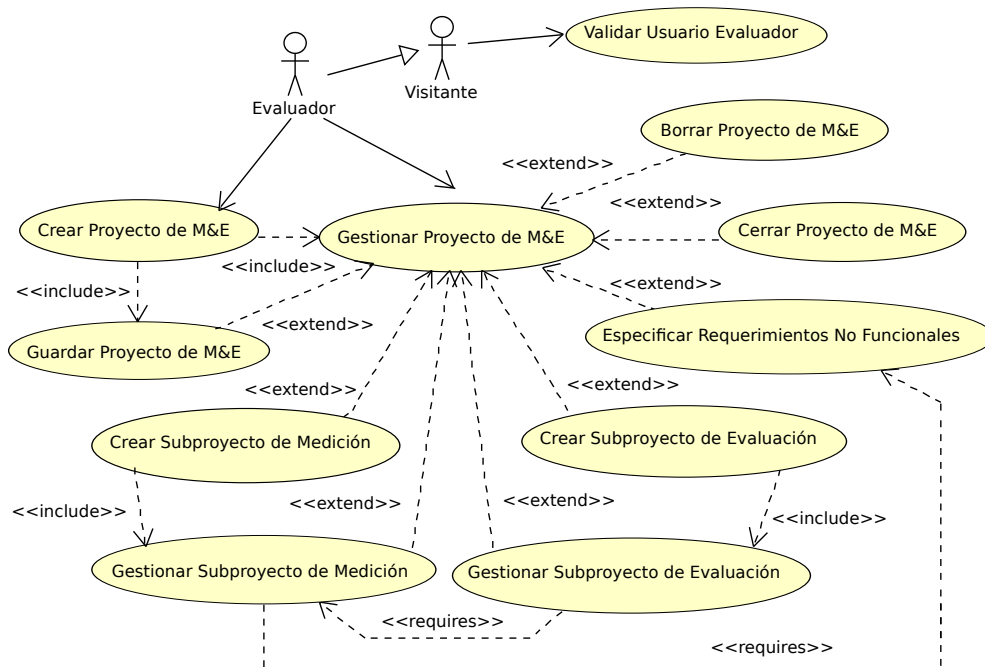
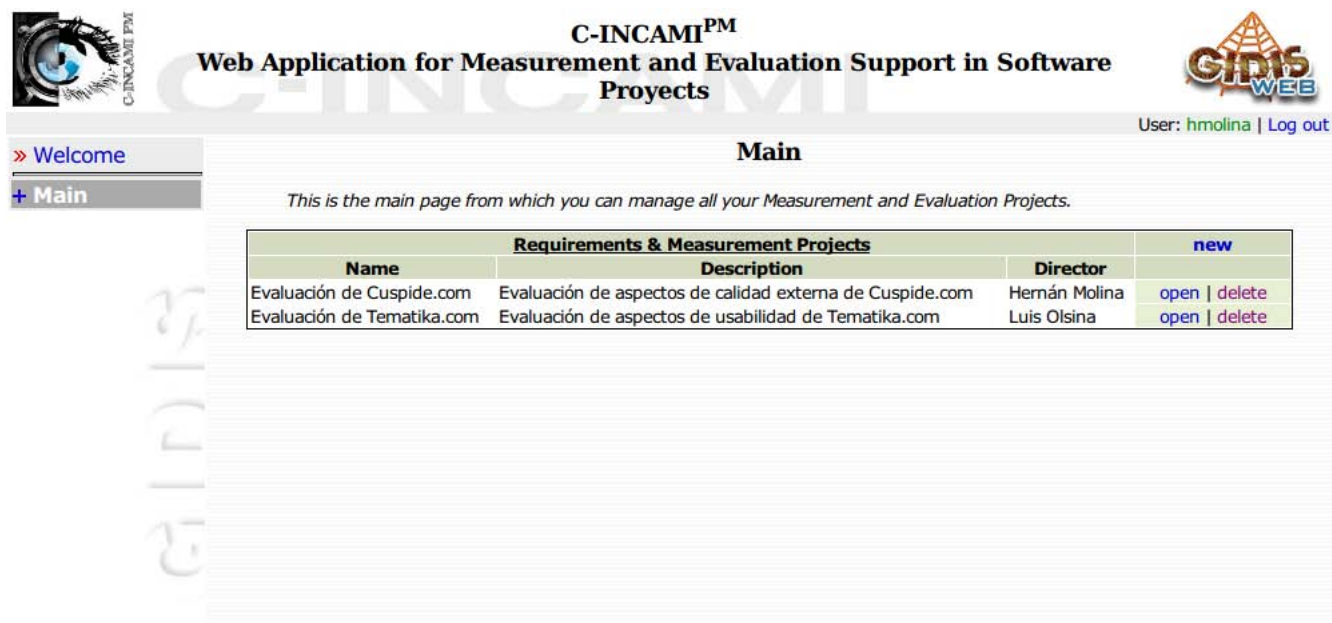


Figura 7.5: Diagrama de Casos de Uso de la herramienta C-INCAMI^{PM} para la Gestión de Proyectos de M&E.

Especificación de Requerimientos No Funcionales Mediante este módulo, la herramienta permite al evaluador especificar la necesidad de información (*InformationNeed*) para un proyecto de M&E (ver Figura 7.8), que incluye:

1. identificar el propósito de la evaluación (*purpose*) y el punto de vista (*userViewpoint*) desde el cual se consideran los requerimientos no funcionales;
2. identificar la categoría de entidad (*EntityCategory*) a la que pertenecen las entidades a ser evaluadas, a partir de las especificadas en el catálogo de M&E. La implementación de este prototipo sólo permite evaluar entidades de una sola categoría, aunque el modelo visto en el Capítulo 4 permite especificar una cantidad arbitraria de las mismas, relacionadas entre sí;

³Se ha definido un estereotipo UML `<<requires>>` para caracterizar esta dependencia.



C-INCAMI^{PM}
Web Application for Measurement and Evaluation Support in Software Projects

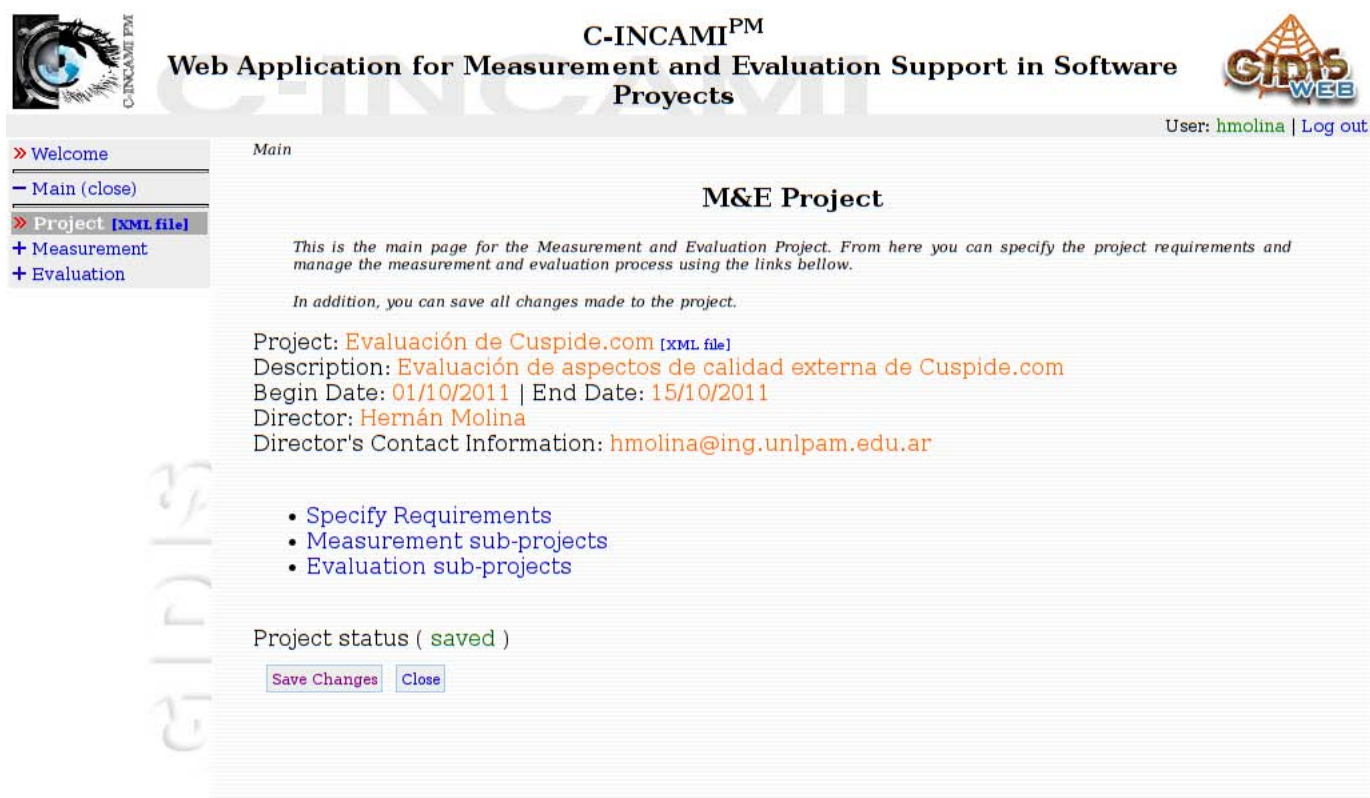
User: [hmolina](#) | [Log out](#)

Main

This is the main page from which you can manage all your Measurement and Evaluation Projects.

Requirements & Measurement Projects			new
Name	Description	Director	
Evaluación de Cuspide.com	Evaluación de aspectos de calidad externa de Cuspide.com	Hernán Molina	open delete
Evaluación de Tematika.com	Evaluación de aspectos de usabilidad de Tematika.com	Luis Olsina	open delete

Figura 7.6: Pantalla de la herramienta C-INCAMI^{PM} que permite crear, editar y borrar Proyectos de M&E.



C-INCAMI^{PM}
Web Application for Measurement and Evaluation Support in Software Projects

User: [hmolina](#) | [Log out](#)

M&E Project

This is the main page for the Measurement and Evaluation Project. From here you can specify the project requirements and manage the measurement and evaluation process using the links below.

In addition, you can save all changes made to the project.

Project: [Evaluación de Cuspide.com](#) [XML file]
 Description: [Evaluación de aspectos de calidad externa de Cuspide.com](#)
 Begin Date: [01/10/2011](#) | End Date: [15/10/2011](#)
 Director: [Hernán Molina](#)
 Director's Contact Information: hmolina@ing.unlpam.edu.ar

- [Specify Requirements](#)
- [Measurement sub-projects](#)
- [Evaluation sub-projects](#)

Project status ([saved](#))

[Save Changes](#) [Close](#)

Figura 7.7: Pantalla de la herramienta C-INCAMI^{PM} que permite gestionar un Proyecto de M&E.

3. identificar el o los conceptos (*CalculableConcept*) foco, que constituirán los requerimientos no funcionales de alto nivel en los cuales se enfocará la evaluación. Éstos se seleccionan de los conceptos especificados en el catálogo de M&E;
4. describir el contexto real (*Context*) de la entidad a evaluar, relevante a la necesidad de información especificada –categoría de entidad, propósito, punto de vista y concepto/s foco– (esta funcionalidad se describe luego);

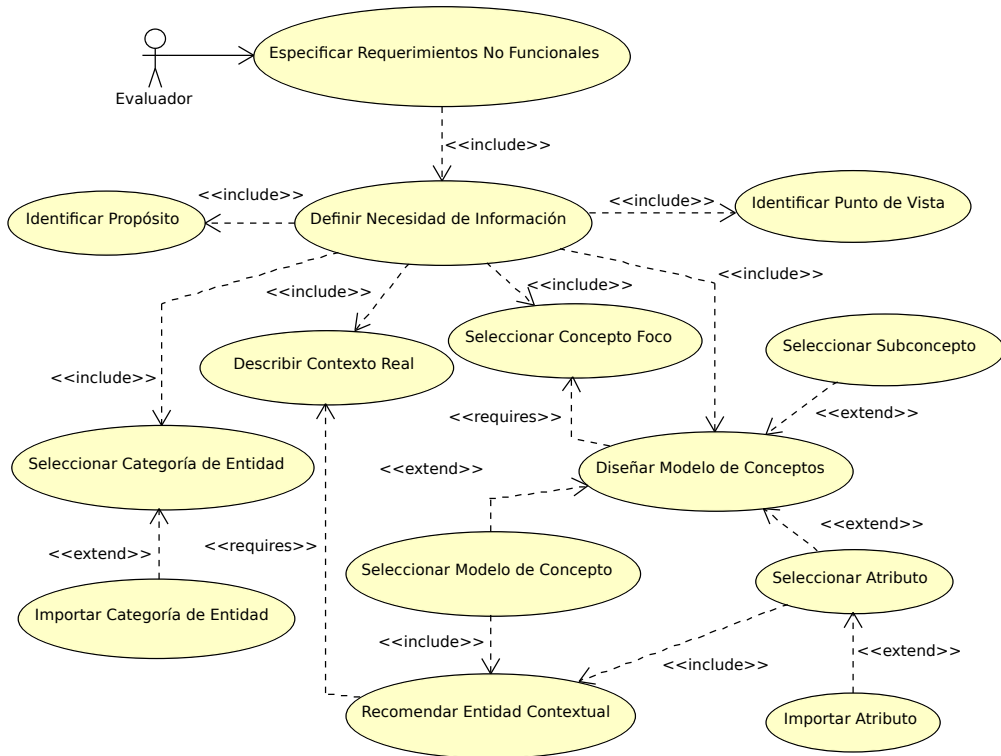
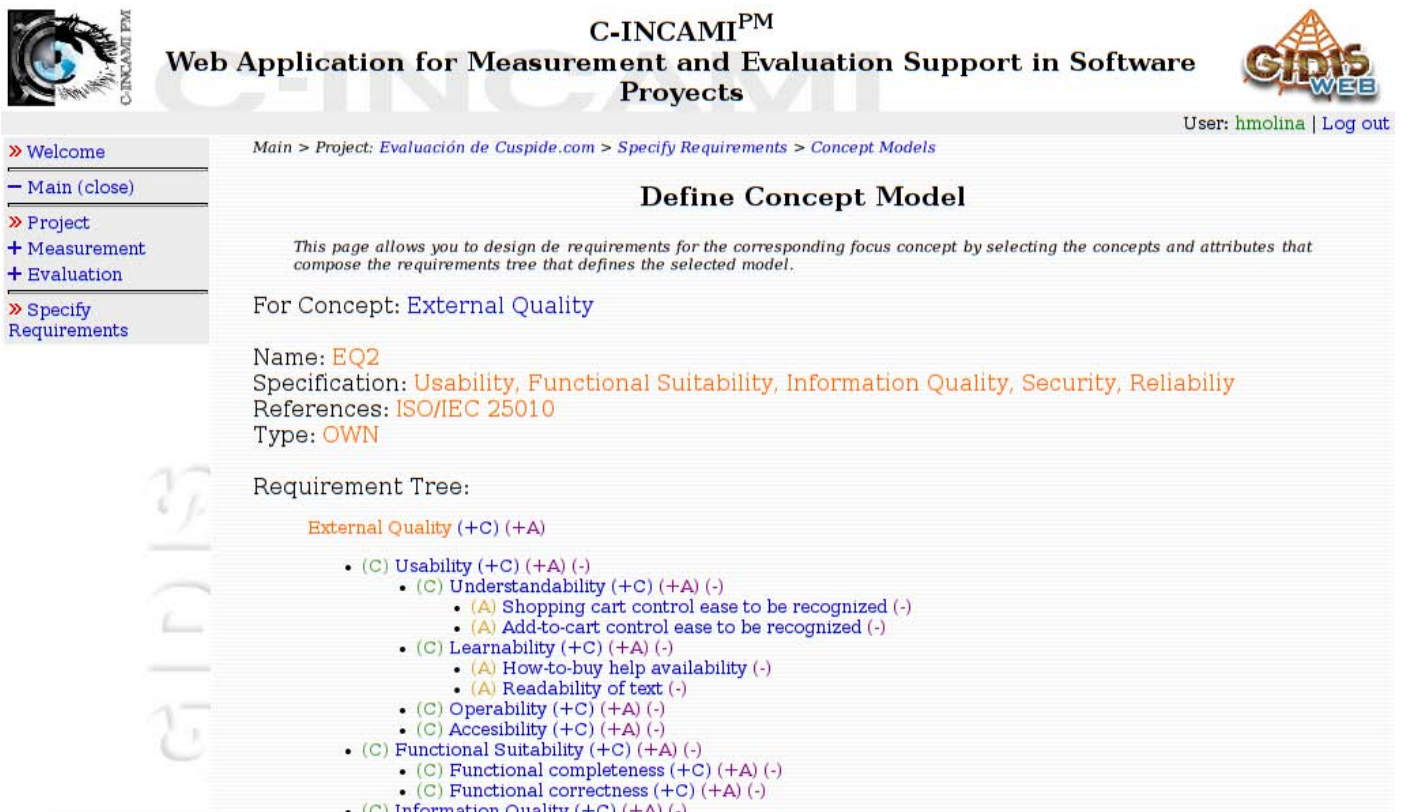


Figura 7.8: Diagrama de Casos de Uso de la herramienta C-INCAMI^{PM} para la Especificación de Requerimientos No Funcionales.

5. seleccionar del catálogo de M&E y/o diseñar un modelo de concepto (*ConceptModel*) que defina los requerimientos no funcionales para cada uno de los conceptos foco de la evaluación. La herramienta permite recuperar del catálogo de M&E un modelo de concepto previamente diseñado (incluyendo conceptos y, opcionalmente, atributos) para representar un determinado concepto foco. La selección del modelo de concepto es asistida por el mecanismo de recomendación sensible al contexto propuesto en la Sección 5.2, presentando la lista de los posibles modelos de forma ordenada según el nivel de aplicabilidad contextual al proyecto actual, indicando dicho valor en cada caso. La herramienta permite luego al evaluador editar el árbol de requerimientos recuperado agregando y/o quitando conceptos (*CalculableConcept*) y atributos (*Attribute*) en cada nivel –tal como fue presentado en la Sección 4.3– (en la Figura 7.9 se muestra la pantalla de la herramienta que permite realizar esta tarea). Tanto los atributos como los conceptos calculables de nivel medio agregados al modelo son seleccionados de las definiciones almacenadas en el catálogo de M&E. La selección de los atributos del catálogo de M&E es también asistida por el mecanismo de recomendación sensible al contexto propuesto en la Sección 5.2 (en la Figura 7.10 se muestra la pantalla de la herramienta que permite seleccionar un atributo, para agregar a un modelo de concepto, usando el mecanismo de recomendación contextual). Opcionalmente las definiciones de los atributos pueden ser importadas de los repositorios del dominio de aplicación de la organización. Se debe notar que el uso del mecanismo de recomendación de entidades contextuales (tales como los atributos y los modelos de concepto, según se definió en la Sección 4.3.1) requiere que el contexto real haya sido descrito previamente (según se muestra en la Figura 7.8). El diseño del modelo de concepto se completa (es decir, es medible) cuando todos los elementos hoja del árbol de requerimientos sean atributos.



C-INCAMI^{PM}
Web Application for Measurement and Evaluation Support in Software Projects

User: hmolina | Log out

Main > Project: Evaluación de Cuspide.com > Specify Requirements > Concept Models

Define Concept Model

This page allows you to design de requirements for the corresponding focus concept by selecting the concepts and attributes that compose the requirements tree that defines the selected model.

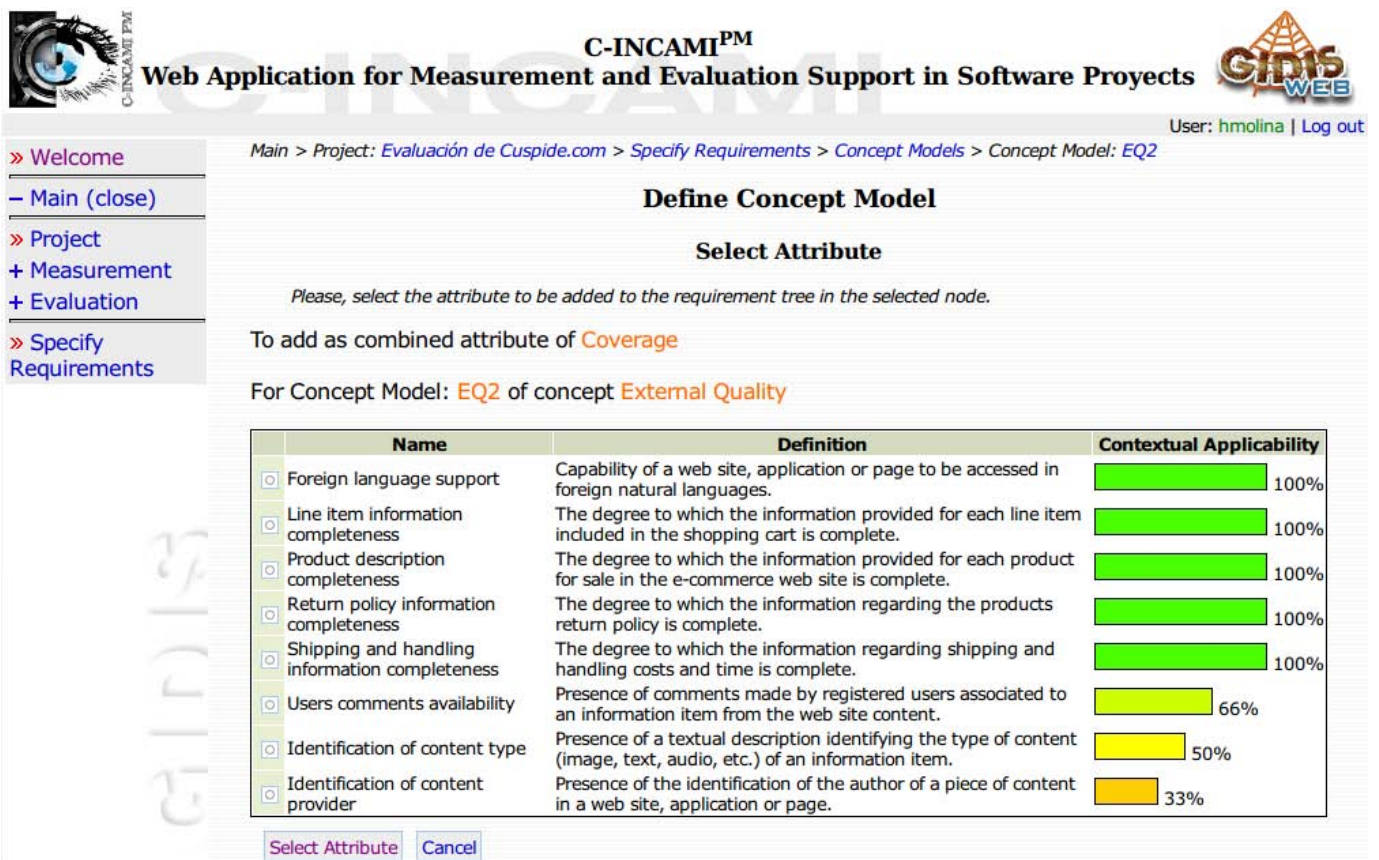
For Concept: External Quality

Name: EQ2
 Specification: Usability, Functional Suitability, Information Quality, Security, Reliability
 References: ISO/IEC 25010
 Type: OWN

Requirement Tree:

- External Quality (+C) (+A)
 - (C) Usability (+C) (+A) (-)
 - (C) Understandability (+C) (+A) (-)
 - (A) Shopping cart control ease to be recognized (-)
 - (A) Add-to-cart control ease to be recognized (-)
 - (C) Learnability (+C) (+A) (-)
 - (A) How-to-buy help availability (-)
 - (A) Readability of text (-)
 - (C) Operability (+C) (+A) (-)
 - (C) Accesibility (+C) (+A) (-)
 - (C) Functional Suitability (+C) (+A) (-)
 - (C) Functional completeness (+C) (+A) (-)
 - (C) Functional correctness (+C) (+A) (-)
 - (C) Information Quality (+C) (+A) (-)

Figura 7.9: Pantalla de la herramienta C-INCAMI^{PM} que permite editar un modelo de concepto.



C-INCAMI^{PM}
Web Application for Measurement and Evaluation Support in Software Projects

User: hmolina | Log out

Main > Project: Evaluación de Cuspide.com > Specify Requirements > Concept Models > Concept Model: EQ2

Define Concept Model

Select Attribute

Please, select the attribute to be added to the requirement tree in the selected node.

To add as combined attribute of Coverage

For Concept Model: EQ2 of concept External Quality

Name	Definition	Contextual Applicability
<input type="checkbox"/> Foreign language support	Capability of a web site, application or page to be accessed in foreign natural languages.	100%
<input type="checkbox"/> Line item information completeness	The degree to which the information provided for each line item included in the shopping cart is complete.	100%
<input type="checkbox"/> Product description completeness	The degree to which the information provided for each product for sale in the e-commerce web site is complete.	100%
<input type="checkbox"/> Return policy information completeness	The degree to which the information regarding the products return policy is complete.	100%
<input type="checkbox"/> Shipping and handling information completeness	The degree to which the information regarding shipping and handling costs and time is complete.	100%
<input type="checkbox"/> Users comments availability	Presence of comments made by registered users associated to an information item from the web site content.	66%
<input type="checkbox"/> Identification of content type	Presence of a textual description identifying the type of content (image, text, audio, etc.) of an information item.	50%
<input type="checkbox"/> Identification of content provider	Presence of the identification of the author of a piece of content in a web site, application or page.	33%

Select Attribute Cancel

Figura 7.10: Pantalla de la herramienta C-INCAMI^{PM} que permite seleccionar un atributo para ser agregado a un modelo de concepto.

Descripción del Contexto Real Como parte de la especificación de los requerimientos no funcionales, la herramienta permite describir el contexto (*Context*) real del proyecto (ver Figura 7.11), que caracteriza a la situación definida por la necesidad de información respecto de la entidad a medir y evaluar. En la Figura 7.12 se muestra la pantalla de la herramienta C-INCAMI^{PM} que muestra las propiedades de contexto –y los valores asignados a cada una– que describen el contexto real del proyecto. Para ello, el evaluador seleccionará del catálogo de M&E las propiedades de contexto (*ContextProperty*) relevantes –pudiendo importar su definición de los repositorios de información de dominio de la organización. Junto a la propiedad, se recupera la especificación de la métrica asignada para cuantificar dicha propiedad. En la Figura 7.13 puede observarse la pantalla de selección de propiedades de contexto en la cual, para asistir al descubrimiento de propiedades relevantes, se presentan las mismas agrupadas en tres vistas diferentes: (i) propiedades asociadas a la entidad evaluada, (ii) propiedades asociadas a las entidades relacionadas y (iii) propiedades relacionadas a las propiedades de contexto ya incluidas. Cuando una propiedad de contexto es seleccionada para ser incluida en la descripción del contexto real, el evaluador debe especificar –como se indicó en la Sección 4.3.1– el peso relativo que la propiedad posee en dicho contexto (*weight*), la relevancia o razón por la cual fue incluida (*relevance*) y la cantidad de valores que pueden ser especificados para la misma (*multiplicityOfValues*). Finalmente, debe especificarse el valor que caracteriza a la propiedad en dicho contexto; para ello, la propiedad de contexto es cuantificada registrando las mediciones correspondientes, obtenidas por la aplicación de la métrica asociada a la propiedad de contexto. En la Figura 7.16 se muestra la pantalla de la herramienta que permite registrar mediciones para un atributo o propiedad a partir de una métrica definida.

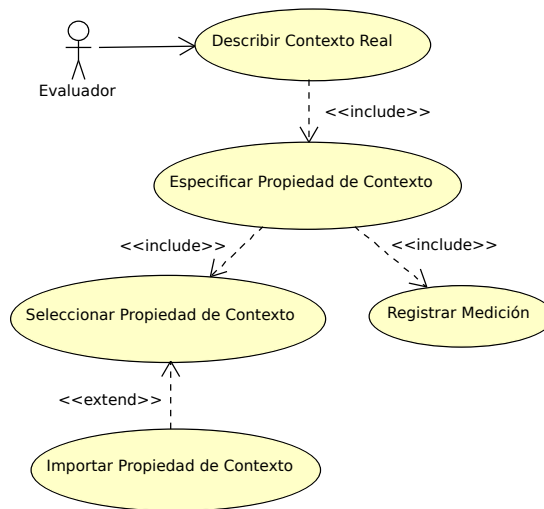
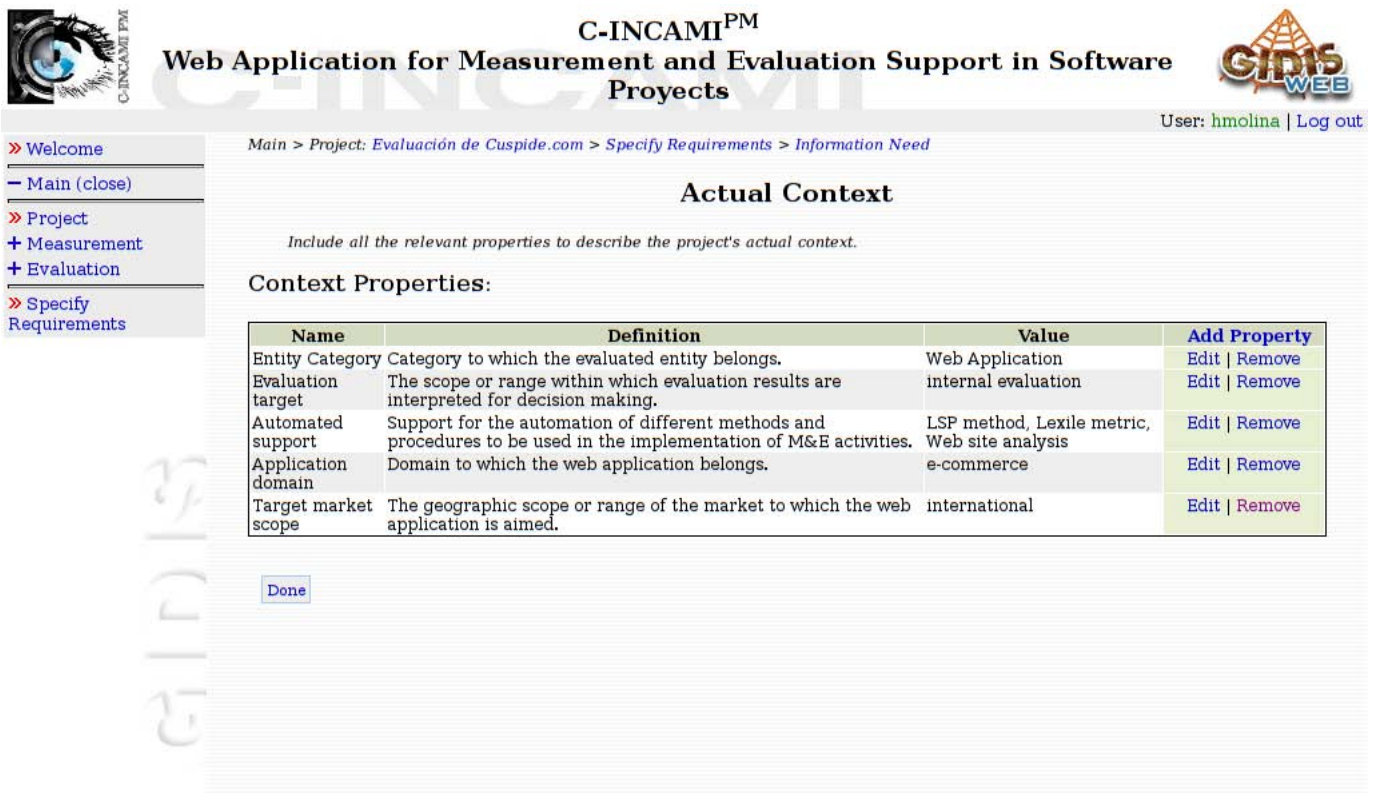


Figura 7.11: Diagrama de Casos de Uso de la herramienta C-INCAMI^{PM} para la Descripción del Contexto Real.

Definición e Implementación de la Medición La implementación de este módulo permite gestionar subproyectos de medición (*MeasurementProject*), todos ellos guiados por la misma necesidad de información especificada inicialmente. La herramienta permite llevar a cabo esta gestión completando tres pasos o tareas (ver Figura 7.14):

1. *Especificación de entidades:* se identifican y especifican las entidades concretas que serán medidas y evaluadas (la herramienta permite especificar más de una), completando los metadatos del concepto correspondiente del marco C-INCAMI (*Entity*). Como se indicó



C-INCAMI^{PM}
Web Application for Measurement and Evaluation Support in Software Projects

User: hmolina | Log out

Main > Project: Evaluación de Cuspide.com > Specify Requirements > Information Need

Actual Context

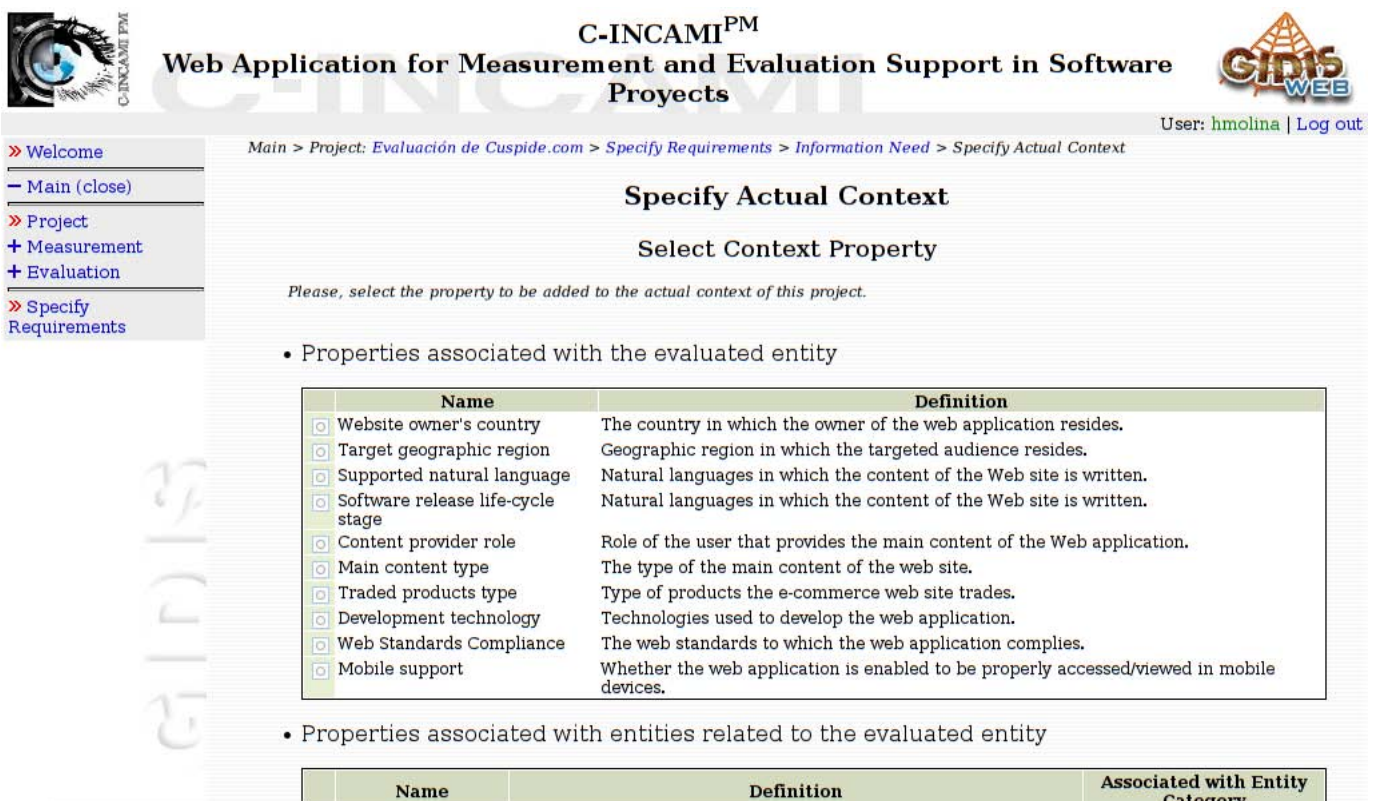
Include all the relevant properties to describe the project's actual context.

Context Properties:

Name	Definition	Value	Add Property
Entity Category	Category to which the evaluated entity belongs.	Web Application	Edit Remove
Evaluation target	The scope or range within which evaluation results are interpreted for decision making.	internal evaluation	Edit Remove
Automated support	Support for the automation of different methods and procedures to be used in the implementation of M&E activities.	LSP method, Lexile metric, Web site analysis	Edit Remove
Application domain	Domain to which the web application belongs.	e-commerce	Edit Remove
Target market scope	The geographic scope or range of the market to which the web application is aimed.	international	Edit Remove

Done

Figura 7.12: Pantalla de la herramienta C-INCAMI^{PM} que muestra las propiedades de contexto incluidas en la descripción del *contexto real*.



C-INCAMI^{PM}
Web Application for Measurement and Evaluation Support in Software Projects

User: hmolina | Log out

Main > Project: Evaluación de Cuspide.com > Specify Requirements > Information Need > Specify Actual Context

Specify Actual Context

Select Context Property

Please, select the property to be added to the actual context of this project.

- Properties associated with the evaluated entity

Name	Definition
<input type="checkbox"/> Website owner's country	The country in which the owner of the web application resides.
<input type="checkbox"/> Target geographic region	Geographic region in which the targeted audience resides.
<input type="checkbox"/> Supported natural language	Natural languages in which the content of the Web site is written.
<input type="checkbox"/> Software release life-cycle stage	Natural languages in which the content of the Web site is written.
<input type="checkbox"/> Content provider role	Role of the user that provides the main content of the Web application.
<input type="checkbox"/> Main content type	The type of the main content of the web site.
<input type="checkbox"/> Traded products type	Type of products the e-commerce web site trades.
<input type="checkbox"/> Development technology	Technologies used to develop the web application.
<input type="checkbox"/> Web Standards Compliance	The web standards to which the web application complies.
<input type="checkbox"/> Mobile support	Whether the web application is enabled to be properly accessed/viewed in mobile devices.

- Properties associated with entities related to the evaluated entity

Name	Definition	Associated with Entity Category
------	------------	---------------------------------

Figura 7.13: Pantalla de la herramienta C-INCAMI^{PM} que permite seleccionar una propiedad de contexto para agregar a la descripción del *contexto real*.

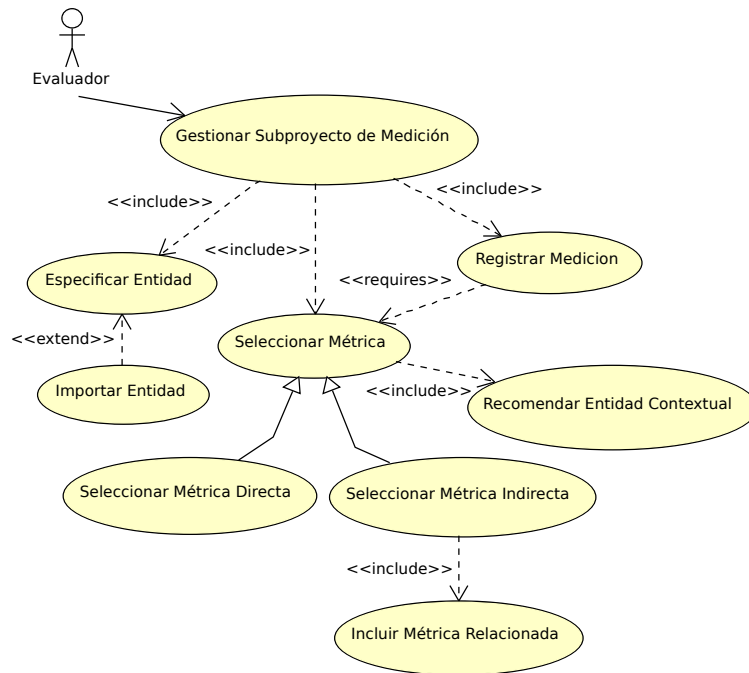


Figura 7.14: Diagrama de Casos de Uso de la herramienta C-INCAMI^{PM} para la Gestión de Proyectos de Medición.

en la Sección 4.6, esta especificación puede ser importada de los repositorios del dominio de aplicación de la organización. Éstas entidades pertenecerán a la categoría de entidad identificada durante la especificación de la necesidad de información.

2. *Selección de métricas:* para cada uno de los atributos incluidos en cada modelo de concepto diseñado durante la especificación de los requerimientos no funcionales, la herramienta permite asignar una métrica, directa (*DirectMetric*) o indirecta (*IndirectMetric*), que será recuperada del catálogo de M&E (en la Figura 7.15 se muestra la pantalla de la herramienta que permite asignar una métrica para cada uno de los atributos a cuantificar para el modelo de concepto seleccionado). Para asistir al evaluador en esta selección, la herramienta presenta el listado de las posibles métricas a aplicar –es decir, aquellas que puedan cuantificar el atributo correspondiente– (cuando fueran más de una) ordenadas por nivel de aplicabilidad al contexto real, indicando a la vez dicho valor (de forma análoga a como se presentaron los atributos en la Figura 7.10). Si el evaluador selecciona una métrica indirecta, las métricas relacionadas (y los atributos cuantificados por éstas últimas) serán también incorporados al proyecto de M&E (de forma recursiva, si las métricas incorporadas fueran también indirectas). La métrica seleccionada será incorporada al proyecto junto a todos los elementos que la definen –la función (si fuera una métrica indirecta), la escala, el método (de medición o de cálculo, según corresponda) y la especificación de la herramienta que la automatiza (si existiera).
3. *Registro de mediciones:* una vez que a un atributo le fue asignada una métrica, la herramienta permite registrar las mediciones correspondientes al mismo, para cada una de las entidades identificadas, en función de la especificación de la métrica asignada (en la Figura 7.16 se muestra la pantalla de la herramienta que permite registrar las mediciones para una métrica/atributo dado). Se deben medir primero todos los atributos cuantificados por métricas directas, incluso aquellos no incluidos en un modelo de concepto –que fueron agregados indirectamente por la selección de una métrica indirecta para cuantificar un atributo del modelo. Luego se miden los atributos del modelo cuantificados por métricas indirectas,

para las cuales la herramienta permite realizar el cálculo asociado de manera automática (si la función correspondiente fue especificada correctamente) seleccionando los valores medidos de las métricas relacionadas usados como entrada. Para cada medición se especifican los metadatos indicados para dicho concepto (*Measurement*) y se asocian a la entidad correspondiente, tal como se definió en la Sección 4.4. Así, cuantificando todos los atributos de cada uno de los modelos de concepto diseñados, para todas las entidades identificadas, se completa la implementación de la medición.

The screenshot shows the C-INCAMI PM web application interface. At the top, there is a navigation bar with the logo 'C-INCAMI PM' and the title 'Web Application for Measurement and Evaluation Support in Software Projects'. On the right, there is a user login area with 'User: hmolina | Log out'. Below the navigation bar, there is a breadcrumb trail: 'Main > Project: Evaluación de Cuspide.com > Measurement Projects > Evaluación de Cuspide.com (medición) > Metrics Selection (Concept Model)'. The main content area is titled 'Measurement Design' and 'Select Attribute'. It contains a message: 'Each attribute in the selected model must be assigned a metric in order to be able to quantify it.' Below this, it states 'For Calculable Concept: External Quality' and 'Modeled by: EQ2'. A section titled 'Attributes in the model:' lists several attributes with their corresponding metrics and actions:

- Foreign Language Support - Quantified by: M Supported Natural Languages [Re-Assign Metric]
- Shipping and handling information completeness - Quantified by: M Available shipping and handling information] [Re-Assign Metric]
- Line item information completeness - Quantified by: M Degree of completeness to the line item information (v2.0)] [Re-Assign Metric]
- Readability of Text - Quantified by: (not specified) [Assign Metric]
- How-to-buy help availability - Quantified by: (not specified) [Assign Metric]
- Shopping cart control ease to be recognized - Quantified by: (not specified) [Assign Metric]

Figura 7.15: Pantalla de la herramienta C-INCAMI^{PM} que permite asignar una métrica a cada atributo de un modelo de concepto.

Definición e Implementación de la Evaluación La implementación de este módulo permite gestionar subproyectos de evaluación. Cada proyecto de evaluación se basa en los resultados de un subproyecto de medición seleccionado, siguiendo la guía establecida por la necesidad de información del proyecto de M&E principal. Para ello, la herramienta permite, para cada uno de los conceptos calculables foco de la evaluación –especificados en la necesidad de información- llevar a cabo los siguientes pasos o tareas (ver Figura 7.17):

1. *Especificación de criterios comunes:* se selecciona del catálogo de M&E una escala (*Scale*) que será utilizada para representar los valores de todos los indicadores que interpretarán los componentes del modelo de concepto diseñado para el concepto foco seleccionado. En caso de tratarse de una escala numérica, se selecciona además, también del catálogo de M&E, la unidad (*Unit*) en la que se expresarán los valores correspondientes. También se selecciona del catálogo de M&E un modelo global (*GlobalModel*) que especifique el mecanismo de agregación de los valores de indicadores elementales y parciales (globales de nivel intermedio) para el cálculo de la evaluación final. En esta selección, la herramienta utiliza el mecanismo de recomendación sensible al contexto para presentar una lista de los

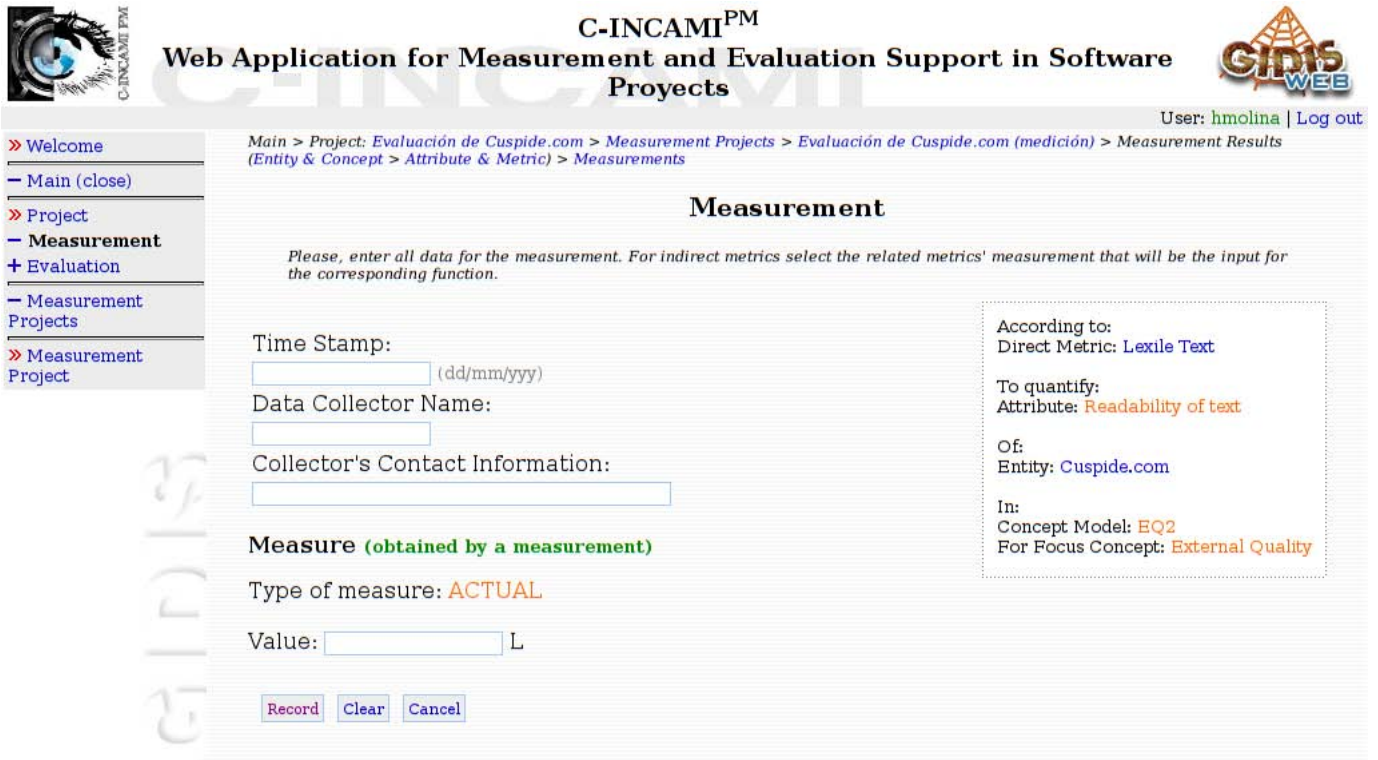


Figura 7.16: Pantalla de la herramienta C-INCAMI^{PM} que permite registrar una medición para una determinada métrica.

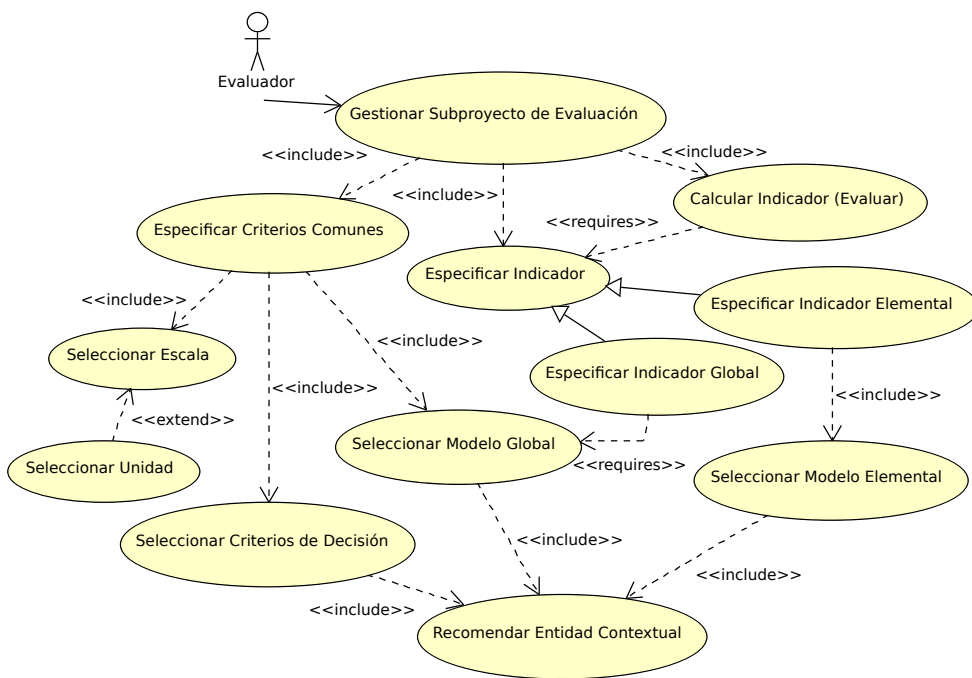


Figura 7.17: Diagrama de Casos de Uso de la herramienta C-INCAMI^{PM} para la Gestión de Proyectos de Evaluación

modelos globales disponibles ordenados por su aplicabilidad al contexto real del proyecto. La implementación actual de la herramienta soporta dos modelos de agregación: el *Modelo LSP* y el *Modelo Aditivo* (este último, utilizando el *Modelo LSP* aplicando el operador *A*, tal como se indicó en la Sección 3.2.3.1). Finalmente, la herramienta permite seleccionar un conjunto de criterios de decisión (*DecisionCriteria*) mediante el cual se proporciona una

interpretación semántica a los valores de indicador resultantes. Nuevamente, la selección de estos criterios es asistida por el mecanismo de recomendación sensible al contexto presentando una lista de los criterios disponibles en el catálogo de M&E ordenados por su nivel de aplicabilidad al contexto real del proyecto actual. En la Figura 7.18 se muestra la pantalla de la herramienta que permite especificar los criterios comunes de evaluación.

2. *Diseño de Indicadores Globales y Elementales*: Una vez que los criterios comunes de evaluación han sido especificados, la herramienta permite diseñar los indicadores que interpretarán los componentes del modelo de concepto para el concepto seleccionado. A partir de un árbol de indicadores, análogo al modelo de concepto correspondiente –donde, recordando la Figura 4.7, cada indicador elemental (*ElementaryIndicator*) evalúa un atributo y cada indicador global (*GlobalIndicator*) evalúa un concepto–, el evaluador podrá diseñar cada uno de los indicadores elementales y globales. Por un lado, para cada indicador elemental, el evaluador seleccionará del catálogo de M&E un modelo elemental (*ElementaryModel*) que especifique cómo convertir los valores de la métrica que cuantifica el atributo interpretado a valores de indicador. Esta selección será respaldada por el mecanismo de recomendación sensible al contexto presentando una lista de los modelos elementales disponibles (aquellos que interpretan la métrica correspondiente) ordenada por su nivel de aplicabilidad al contexto real del proyecto actual. Así mismo, el evaluador deberá asignar el peso del indicador relativo al resto de los indicadores agregados en el mismo nivel del árbol (que será utilizado por el modelo global seleccionado para la agregación de valores de indicadores). En la Figura 7.19 se muestra la pantalla de la herramienta que permite diseñar un indicador elemental. Por otro lado, para cada indicador global el evaluador asignará los parámetros correspondientes al modelo global elegido (previamente seleccionado y común a todos los indicadores globales). En primer lugar, corresponde asignar el operador lógico utilizado para calcular la agregación de los valores de los indicadores relacionados (por ejemplo, alguno de los operadores definidos por el Modelo LSP –presentados en la Figura 3.4–, si este fuera el modelo global elegido). En segundo lugar, se asigna el peso del indicador relativo a los indicadores agregados en el mismo nivel.
3. *Evaluación de Entidades*: Finalmente, para cada entidad identificada durante la gestión del subproyecto de medición asociado, se lleva a cabo la evaluación del concepto foco seleccionado, utilizando los indicadores diseñados en el paso anterior. Para ello, primero es necesario indicar, para cada uno de los indicadores elementales, el valor medido del atributo correspondiente que será utilizado como entrada al modelo elemental (considerando que más de una medición podría haber sido registrada para el mismo atributo). Si sólo una medición se hubiera registrado para tal atributo, la herramienta la selecciona automáticamente. Luego, el evaluador puede ordenar el cálculo de todos los indicadores globales, el cual se lleva a cabo automáticamente para cada uno de los indicadores globales, desde el nivel más bajo hasta el indicador global que interpreta el concepto foco elegido.

Estos pasos deben ser repetidos para todas las entidades medidas y para todos los conceptos calculables que fueron especificados como foco de la evaluación durante la especificación de la necesidad de información. Los resultados de la evaluación para cada entidad y cada concepto foco pueden ser consultados en todo momento. Para cada caso, la herramienta muestra los resultados de la evaluación sobre el modelo de concepto correspondiente, indicando, para cada componente, el valor de indicador resultante utilizando la escala (y unidad) seleccionada más una representación gráfica del mismo, destacando el nivel de aceptabilidad (definido en los criterios de decisión) en el que se encuentra el valor del indicador (se crea una escala de colores entre el verde y el rojo a partir de los niveles definidos en los criterios de decisión), tal como se muestra en la Figura 7.20).

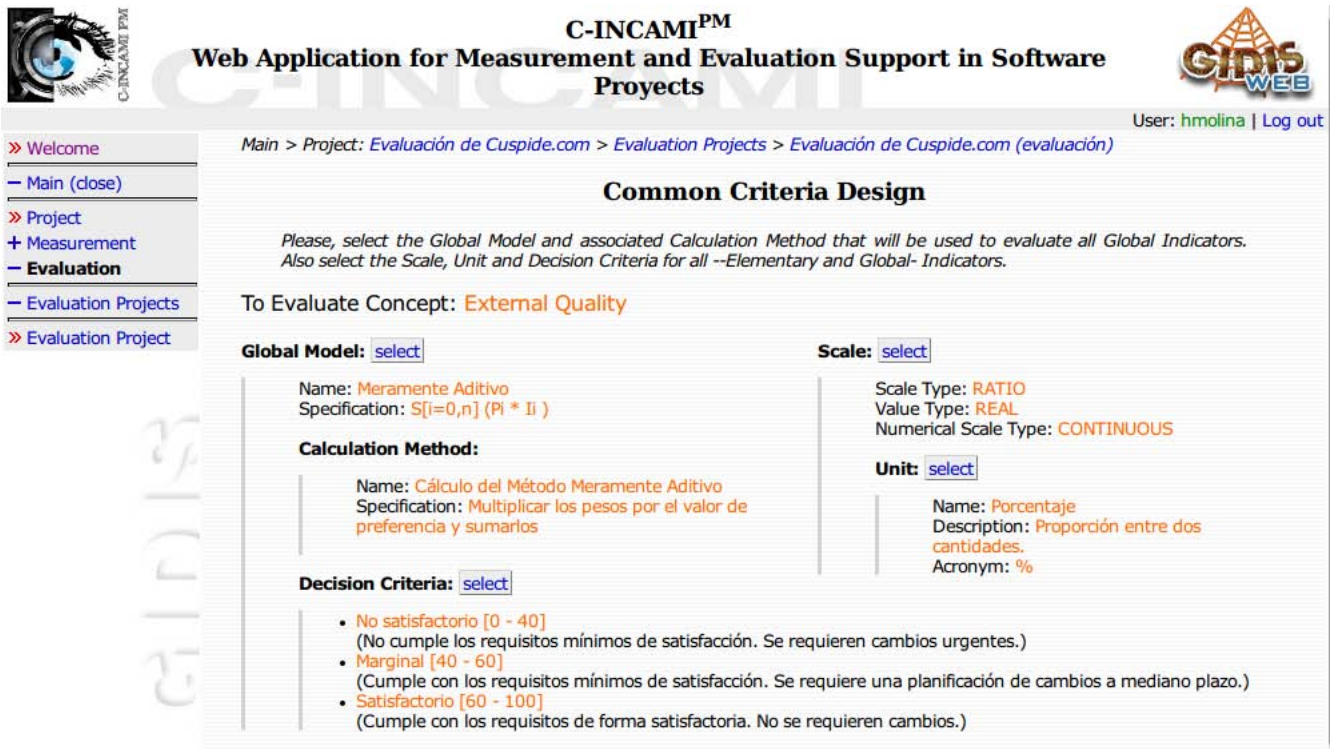


Figura 7.18: Pantalla de la herramienta C-INCAMI^{PM} que permite especificar criterios comunes de evaluación.

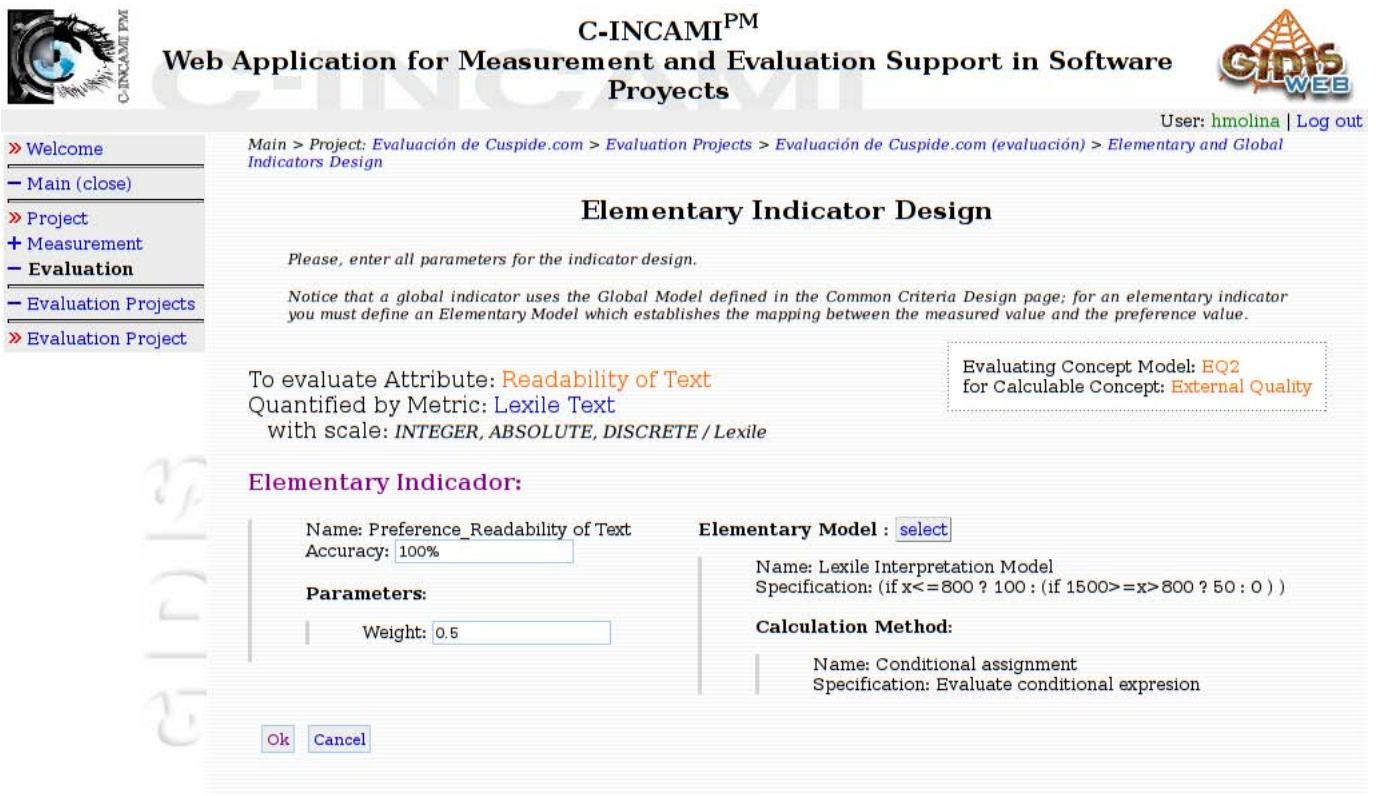


Figura 7.19: Pantalla de la herramienta C-INCAMI^{PM} que permite especificar un indicador elemental.

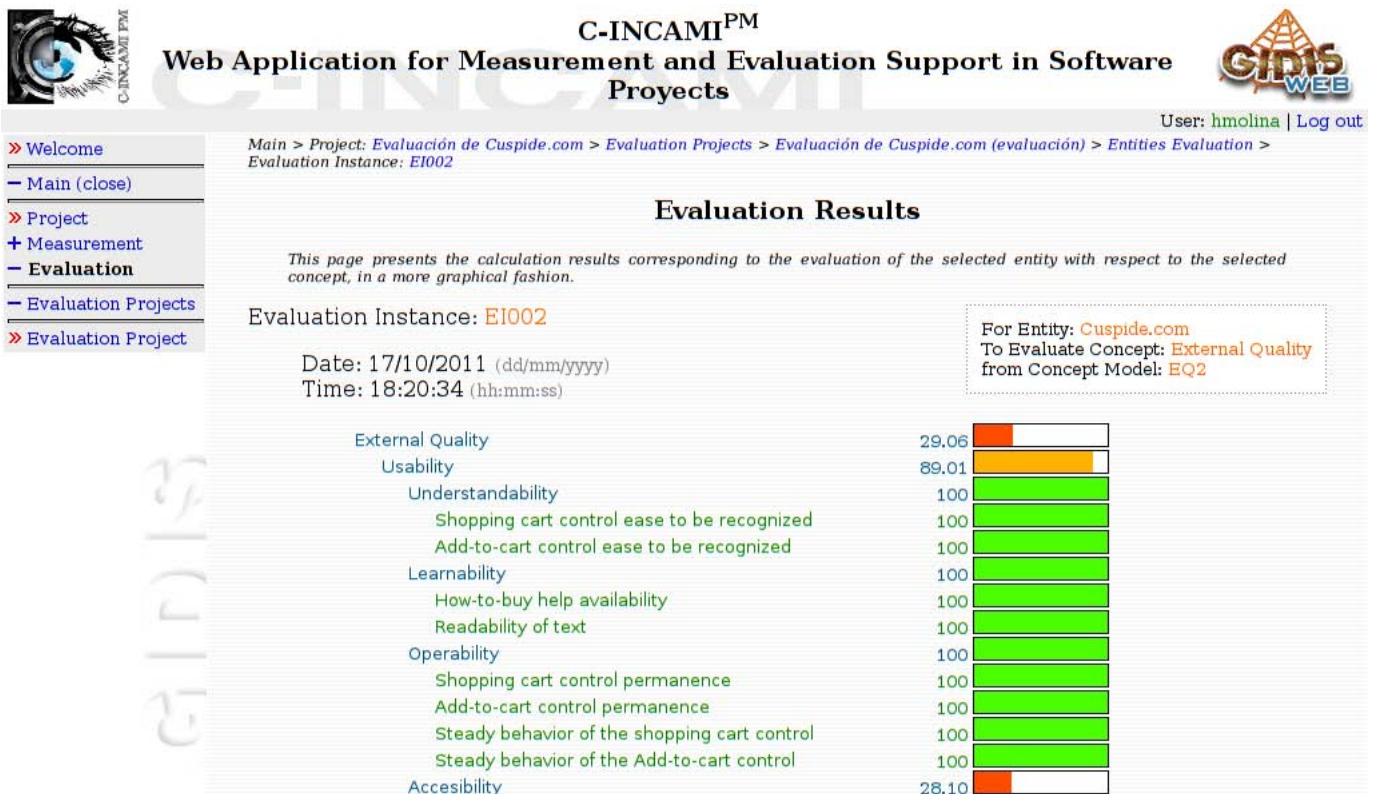


Figura 7.20: Pantalla de la herramienta C-INCAMI^{PM} que muestra los resultados de evaluación para una entidad y concepto dado.

7.2.1.3. Capa de Lógica de Acceso a Datos

La capa de lógica de acceso a datos consta de los componentes que permiten acceder a los servicios de persistencia utilizados, ocultando la complejidad inherente a los mismos y, al mismo tiempo, convirtiendo los tipos de datos leídos y escritos de forma apropiada para permitir su manipulación por parte de los módulos de la lógica de negocio del sistema. En esta capa existen cuatro módulos, cada uno responsable de acceder a cada uno de los repositorios persistentes de información presentados en la Figura 7.1. A continuación se describen aspectos funcionales y de implementación de cada uno de estos módulos.

Módulo de Acceso al Catálogo C-INCAMI Este módulo [Molina *et al.*, 2004] contiene un conjunto de componentes que permiten a la herramienta recuperar las especificaciones de elementos de información reusables almacenadas en el catálogo de M&E (atributos, conceptos, métricas, etc.), ocultando los detalles inherentes a la manipulación de especificaciones RDF en las que se almacenan dichas especificaciones en la capa subyacente de la arquitectura –tal como se indicó en la Sección 3.3.3 y 4.6 y como se verá luego cuando se describa el catálogo de M&E en la Sección 7.2.2.

Al igual que en el módulo descrito anteriormente los componentes de este módulo consisten en paquetes JavaTM, cada uno de ellos agrupando un conjunto de clases relacionadas. El paquete principal, `gidisweb.c_incami.repository`, es responsable de establecer la conexión y efectuar consultas al catálogo de M&E. Adicionalmente, un conjunto de subpaquetes son responsables de recuperar representaciones computables (objetos JavaTM) a partir de las instancias de datos almacenadas en el catálogo para cada uno de los conceptos descritos en el Capítulo 4. Estos subpaquetes corresponden a los diferentes módulos del marco C-INCAMI, a saber:

- `gidisweb.c_incami.repository.requirements`

- gidisweb.c_incami.repository.context
- gidisweb.c_incami.repository.measurement
- gidisweb.c_incami.repository.evaluation

Cada una de las clases incluidas en estos paquetes actúa como *factory* de objetos Java™, correspondientes a los conceptos reusables del marco C-INCAMI, a partir de las instancias RDF almacenadas en el catálogo de M&E (ver Tabla 7.1). Estas clases permiten además efectuar consultas de búsqueda basadas en las relaciones entre los conceptos, obteniendo como resultado un listado de las instancias disponibles en el catálogo que satisfacen el criterio de búsqueda (por ejemplo, las métricas que cuantifican un determinado atributo). Estos listados son utilizadas en la selección de elementos de información reusables durante las actividades de diseño de la medición y la evaluación (tal como se ilustró en el Capítulo 6).

Tabla 7.1: Clases incluidas en los subpaquetes de `gidisweb.c_incami.repository`.

Requerimientos	Contexto
AttributeFactory CalculableConceptFactory ConceptModelFactory EntityCategoryFactory	ContextPropertyFactory
Medición	Evaluación
MetricFactory ScaleFactory UnitFactory MethodFactory FunctionFactory	ElementaryModelFactory GlobalModelFactory DecisionCriteriaFactory

Tanto las consultas de búsqueda como de extracción de instancias RDF se realizan mediante el lenguaje de consultas RQL (*RDF Query Language*, introducido en la Sección 4.6). Por ejemplo, la clase `AttributeFactory` provee dos métodos para tal fin:

- Hashtable `getAttributeList(String login4, String entCatURI)`, devuelve una tabla indexada (tabla *hash*) conteniendo los nombres y URIs de los atributos asociados a la categoría de entidad indicada, que se encuentran almacenados en el catálogo de M&E, utilizando la siguiente consulta RQL:

```
"SELECT uri, name "+
"FROM {uri} incami:AttributeName {name}; "+
"incami:Associated_with {<"+entCatURI+">} "+
"using namespace "+
"incami = <http://"+C_INCAMIFramework.IP+":8080/incami.rdf#>";
```

- `Attribute getAttribute(String uri, String login, EntityCategory entityCategory)`, devuelve el atributo (objeto *Attribute*) indicado (mediante su *URI*) a partir de su definición almacenada en el catálogo de M&E, utilizando la siguiente consulta RQL:

⁴El identificador del usuario permite controlar que la consulta sea efectuada por un usuario validado por la herramienta.

```
"CONSTRUCT {sub} pred {obj} "+
"FROM {sub} pred {obj}, "+
"{sub} rdf:type {incami:Attribute} "+
"WHERE sub = <"+uri+"> "+
"using namespace "+
"incami = <http://"+C_INCAMIFramework.IP+":8080/incami.rdf#>"
```

Se debe notar que la primer consulta utiliza el comando *SELECT* que devuelve el resultado como una tabla, similar a la devuelta por una consulta SQL (estructurada en filas y columnas), a partir de la cual se instancia la tabla *hash* devuelta por el método. Por otro lado, en la segunda consulta se utiliza el comando *CONSTRUCT* que devuelve el resultado en la forma de un grafo RDF, compuesto por un conjunto de sentencias (objetos de la clase `org.openrdf.model.Statement`) a partir de las cuales pueden obtenerse sus componentes utilizando los métodos apropiados `getSubject()`, `getPredicate()` y `getObject()` provistos por el API correspondiente a la tecnología utilizada Sesame-API [2006]. A partir del grafo obtenido como resultado, el método instancia un objeto *Attribute* (del paquete `gidisweb.c_incami.requirements`) que es devuelto como resultado a la invocación del método `getAttribute`.

Módulo de Acceso a Datos del Dominio de Aplicación Este módulo es responsable de acceder a los repositorios de información existentes en la organización con el objetivo de importar al catálogo de M&E las especificaciones utilizadas en actividades de medición –tales como atributos, categorías de entidad, entidades, unidades, escalas categóricas y valores categóricos (tal como se presentó en la Sección 4.6)- a partir de las definiciones e instancias del dominio de aplicación almacenadas en los repositorios de la organización. Típicamente, estos repositorios estarán implementados en bases de datos relacionales, por lo que este módulo debe recurrir a mecanismos que interpreten esquemas y datos representados mediante el modelo relacional y los conviertan a definiciones de términos e instancias representados mediante el modelo de datos de RDF. Este es el propósito de un importante número de trabajos Berners-Lee [1998]; Sahoo *et al.* [2009]; Bizer *et al.* [2009] incluyendo los presentados en el “*W3C Workshop on RDF Access to Relational Databases*” W3C-RDB2RDF-Workshop [2007], del cual surgió el “*RDB2RDF Working Group*” del W3C W3C-RDB2RDF-Group [2011] dedicado a estandarizar lenguajes para convertir datos y esquemas de bases de datos relacionales en RDF y OWL. De esta línea de investigación han surgido un conjunto de resultados entre los que se encuentran dos borradores que documentan una correspondencia directa de datos relacionales a RDF W3C-RDBDirectMapping [2011] y una propuesta de lenguaje para declarar la conversión entre bases de datos relacionales y RDF W3C-R2RML [2011].

Cabe aclarar que este módulo no fue implementado en el prototipo creado. No obstante, y a la luz de los desarrollos existentes en el área citados arriba, es válido considerar la factibilidad del *módulo de acceso a datos del dominio de aplicación* incorporado a la arquitectura de la herramienta C-INCAMI^{PM}.

Módulo de Control de Acceso Este módulo es responsable de validar el acceso de los usuarios evaluadores a la herramienta accediendo al repositorio de la capa subyacente que almacena la información de los evaluadores registrados en la aplicación así como los proyectos que administra cada uno de ellos. Este módulo, como los anteriores, está implementado en JavaTM y utiliza los conectores apropiados para efectuar consultas SQL al repositorio mencionado que, como se verá luego, está implementado como una base de datos relacional.

El módulo está conformado por un solo paquete, `gidisweb.c_incami.database`, cuya única clase *DataBaseClient* ofrece a los módulos de la capa superior un conjunto de servicios específicos de forma transparente y en función de los conceptos y el vocabulario del marco C-INCAMI:

- `queryUser(String userLogin, String userPassword)`: permite determinar si el usuario con nombre y clave de acceso indicados, es un usuario válido del sistema. En caso afirmativo, devuelve un objeto *RegisteredUser* apropiadamente instanciado, que los módulos de la capa superior utilizarán de forma activa para todas las tareas que el usuario solicite;
- `queryProjects(String userLogin)`: devuelve un listado de los proyectos administrados por un usuario determinado;
- `updateProject(MEProject project, String login)`: actualiza los datos de un proyecto determinado para el usuario indicado;
- `deleteProject(String projectName, String login)`: borra un proyecto determinado para el usuario indicado.

Módulo de Acceso a Datos de Proyectos Este módulo es responsable de gestionar (almacenar y recuperar) los resultados del diseño e implementación llevados a cabo en proyectos de M&E en repositorios persistentes diseñados para tal fin. Esta responsabilidad requiere que las especificaciones y resultados de proyectos de M&E puedan ser convertidas a un formato serializable, es decir, un formato que permite almacenar de forma persistente (o transmitir por una conexión remota) un conjunto de datos estructurados para luego ser recreados en su forma original y ser manipulados por el mismo u otro sistema en ejecución. El formato seleccionado para tal propósito es XML ya que permite representar de forma consistente (sin pérdida de información y manteniendo todos los aspectos estructurales) la estructura de los datos y metadatos de M&E manipulados por la herramienta (aquella definida por los módulos del subsistema C-INCAMI –presentado en la Sección 7.2.1.2). Por lo tanto para implementar este módulo se requiere un mecanismo que permita convertir todos los objetos (y sus asociaciones) manipulados por la herramienta, estructurados según el subsistema C-INCAMI, a XML (para su almacenamiento) y de vuelta a los objetos (y asociaciones) originales. Para realizar esta transición, se recurrió a la librería de código abierto Castor⁵, implementada en Java™, que permite realizar la conversión de un modelo de objetos hacia XML y viceversa (proceso conocido como *XML Data Binding*). Esta librería permite, por un lado, especificar un archivo XML que define las reglas de conversión a XML para modelos de objetos existentes; y por otro lado, la capacidad de soportar el enlace entre Java™y XML sin poseer un esquema XML que defina cada elemento del documento.

Para establecer claramente la forma en que se convierten los objetos manipulados por la herramienta C-INCAMI^{PM} a XML se definieron las reglas de conversión entre el modelo de objetos existente (la implementación del subsistema C-INCAMI) y el formato deseado del documento XML donde se almacenarían los proyectos de M&E. Estas reglas se especifican en un documento XML (`incami-mapping.xml`) con un esquema definido que establece los elementos necesarios para declarar la conversión para cada clase. Básicamente, las reglas de conversión para una clase se declaran con una etiqueta `<class>` conteniendo uno o varios elementos `<field>` que establecen cómo se convertirán los atributos de la clase. Por ejemplo, las reglas de conversión para la clase *InformationNeed* son las siguientes:

⁵Framework de código abierto para Java que provee enlace Java-a-XML y persistencia Java-a-SQL (JDO –Java Data Objects) [<http://www.castor.org>]

```

<class name="c_incami.requirements.InformationNeed">
  <field name="purpose" type="string"/>
  <field name="viewpoint" type="string"/>
  <field name="actualContext" type="c_incami.context.Context"
    set-method="setActualContext" get-method="getActualContext"/>
  <field name="focusConcepts" type="c_incami.requirements.CalculableConcept"
    collection="hashtable" set-method="addFocusConcept"
    get-method="getFocusConcepts">
    <bind-xml reference="true" node="element"/>
  </field>
  <field name="object" type="c_incami.requirements.EntityCategory"
    set-method="setEntityCategory"
    get-method="getEntityCategory"/>
</class>

```

Entonces para un objeto *InformationNeed* en java, representado en diagrama de la Figura 7.21 se tendría la siguiente representación en XML:

```

<information-need>
  <purpose>Mejorar</purpose>
  <viewpoint>Usuario Final</viewpoint>
  <actual-context name="Actual Context"
    description="Actual context description for the M&E Project">
    <context-property name="..."> ... </context-property>
    ...
  </actual-context>
  <focus-concepts>Eficiencia</focus-concepts>
  <object name="Grupo de Desarrollo">
    <description>Conjunto de personas con diferentes roles que llevan a cabo
      un proyecto de software </description>
    <super-category>Recurso</super-category>
  </object>
</information-need>

```

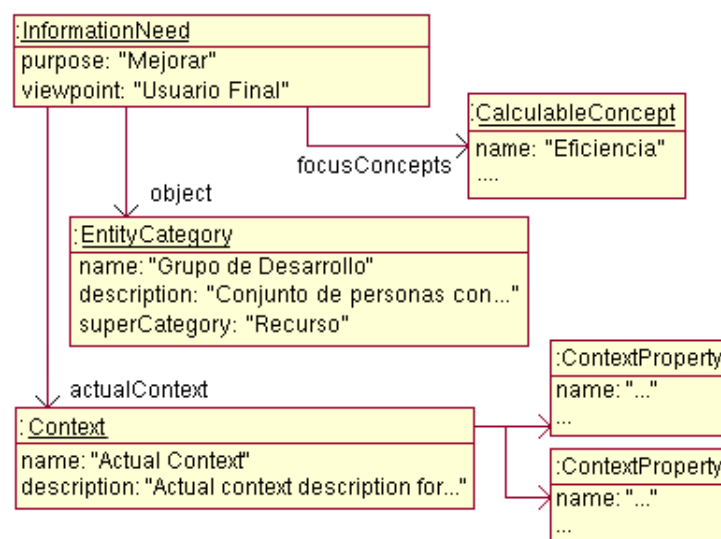


Figura 7.21: Representación en UML de los objetos instanciados para especificar una necesidad de información.

El paquete `gidisweb.c_incami.xml` es el que conforma el módulo de acceso a datos de proyectos, conteniendo una única clase, *IncamiXMLBinding*, que permite realizar –haciendo uso de la librería Castor- la *serialización* de los datos completos de un proyecto (un objeto de la clase *MEProject*) a XML y la *deserialización* correspondiente para recrear el mismo objeto original. De esta forma, los datos XML de un proyecto de M&E pueden ser guardados en archivos de texto (.xml) en el servidor de aplicación. Básicamente, la clase *IncamiXMLBinding* posee dos métodos para llevar a cabo dicha funcionalidad:

- `boolean write(MEProject project)`: toma como argumento un objeto *MEProject*, a partir del cual pueden accederse a las especificaciones y resultados completos de M&E, lo *serializa* a XML y lo almacena de forma persistente en un archivo .xml en un espacio del disco del servidor reservado para el usuario evaluador responsable.
- `MEProject read(String projectName)`: toma como argumento el nombre de un proyecto de M&E serializado y almacenado previamente en el servidor en un archivo .xml y *deserializa* el contenido del mismo recreando el objeto *MEProject* original, conteniendo todas las especificaciones y resultados completos de M&E.

Como lo indica el diagrama de la arquitectura de la herramienta presentado en la Figura 7.1 este módulo es utilizado de forma transparente por el submódulo de Administración de Proyectos de M&E del subsistema C-INCAMI para proveer el servicio de almacenamiento y recuperación de proyectos de M&E.

7.2.1.4. Capa de Persistencia

La capa de persistencia consta de los repositorios utilizados por la herramienta para almacenar tanto los datos de los usuarios evaluadores registrados de la organización, como los datos y metadatos de los proyectos de M&E de la misma. Cada repositorio utiliza diferentes tecnologías para cubrir los requerimientos particulares del propósito de cada uno.

Como ya se adelantó en la Sección 7.2.1.3 la herramienta gestiona dos repositorios propios:

- Un repositorio de datos de los usuarios evaluadores y los proyectos de M&E gestionados por ellos, implementado como una base de datos relacional, y
- Un repositorio de memorias de proyectos de M&E que registran todos los metadatos y datos resultantes del diseño e implementación de M&E en cada uno de los proyectos, implementado como un conjunto de archivos de texto cuyo contenido satisface el estándar XML (archivo .xml).

A continuación se describen cada uno de estos repositorios.

Repositorio de Datos de Usuarios Este repositorio almacena los datos de usuarios evaluadores habilitados para gestionar proyectos de M&E así como los nombres de los proyectos gestionados para cada uno de ellos. Considerando el bajo volumen y complejidad que involucra la gestión de estos datos, una base de datos relacional resultó la alternativa más apropiada. De esta forma, el acceso a tales datos se puede realizar de forma simple y rápida. En la Figura 7.22 se muestra el diagrama Entidad-Relación de las tablas de dicha base de datos.

Para implementar esta base de datos se utilizó el sistema de administración de base de datos relacional *MySQL*⁶ que provee una arquitectura cliente/servidor y un conjunto de librerías⁷ que proveen

⁶MySQL Community Server [<http://www.mysql.com/downloads/mysql/>]

⁷MySQL Connectors [<http://dev.mysql.com/downloads/connector/>]

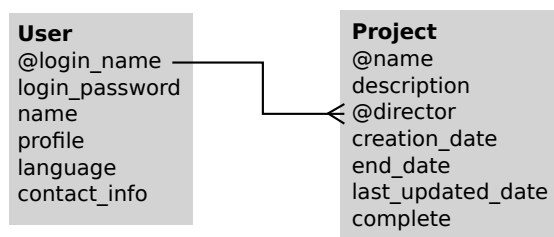


Figura 7.22: Diagrama Entidad-Relación del esquema de la base de datos de usuarios de la herramienta C-INCAMI^{PM}.

conectividad a herramientas compatibles con *ODBC* (*Open Database Connectivity*⁸) y *JDBC* (*Java Database Connectivity*⁹). Particularmente, se utilizó la librería *MySQL Connector/J* que utiliza *JDBC* para permitir el acceso a bases de datos para la plataforma JavaTM.

Como se comentó en la Sección 7.2.1.3, este repositorio es gestionado por el *Módulo de Control de Acceso* para ofrecer al subsistema C-INCAMI un servicio transparente de validación y control de acceso a evaluadores y proyectos de M&E.

Repositorio de Proyectos de M&E Este repositorio almacena los diseños y resultados obtenidos en la realización de los proyectos de M&E de la organización gestionados mediante la herramienta C-INCAMI^{PM}. Dado el volumen y complejidad de los datos utilizados por la herramienta para especificar tal información, se requiere de una tecnología de almacenamiento rápida y fácil de estructurar. También es requerimiento de esta implementación la posibilidad de reutilizar dichos datos en otros sistemas. Por esta razón, se recurre al lenguaje *XML*. Entonces, el repositorio de proyectos de M&E está conformado por documentos *XML*, uno para cada proyecto de medición y evaluación (*MEProject*) creado y gestionado mediante la herramienta C-INCAMI^{PM} y cuya estructura se define a partir de las reglas de conversión Java-XML instrumentadas por la librería *Castor* (descrita en la Sección 7.2.1.3). La herramienta almacena en, y recupera de, cada documento *XML* todos los metadatos y datos que conforman la especificación de requerimientos, diseño e implementación de M&E para las entidades correspondientes. Cada archivo .xml recibe el nombre que le fue dado al proyecto en la base de datos de usuarios de manera que la ubicación del mismo es directa. El prototipo de la herramienta implementada permite además exportar el documento *XML* generado al guardar el proyecto. Como se comentó en la Sección 7.2.1.3, estos documentos son gestionados por el *Módulo de Acceso a Datos de Proyecto* para ofrecer al subsistema C-INCAMI un servicio transparente de persistencia en un formato apropiado.

7.2.2. Catálogo de M&E

El catálogo de M&E almacena especificaciones reusables utilizadas en el diseño de las actividades desarrolladas en los proyectos de M&E, utilizando tecnologías propuestas por la web semántica, particularmente *RDF*. Para implementar este componente se utilizó *Sesame*¹⁰, un repositorio semántico, implementado en JavaTM, que permite almacenar de forma persistente esquemas y datos *RDF* y efectuar consultas a los mismos utilizando inferencia semántica. *Sesame* permite definir repositorios persistentes sobre diferentes tecnologías de almacenamiento tales como bases de datos, archivos *RDF* o memoria principal. El código específico de la tecnología de almacenamiento se concentra en la capa de la arquitectura de *Sesame* llamada *SAIL* (*Storage And Inference Layer*) que ofrece a sus clientes, a

⁸Microsoft Open Database Connectivity (ODBC) [[http://msdn.microsoft.com/en-us/library/windows/desktop/ms710252\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ms710252(v=vs.85).aspx)]

⁹Java SE Technologies - Database [<http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136101.html>]

¹⁰<http://www.openrdf.org/>

través de un API, métodos específicos para la manipulación y consulta de datos y esquemas RDF y los traduce a llamadas a código específico del mecanismo de almacenamiento subyacente. A través de la capa SAIL, *Sesame* permite cargar y eliminar esquemas y datos RDF, evaluar consultas y extraer datos y/o esquemas completos en formato RDF (u otros compatibles). Para efectuar consultas sobre los datos y esquemas RDF, *Sesame* permite utilizar los lenguajes SPARQL [W3C-SPARQL, 2008] o SeRQL –un lenguaje propio que incorpora las características del primero. La arquitectura cliente/servidor de *Sesame* permite que el repositorio pueda ser accedido de forma remota utilizando un API apropiado, permitiendo utilizar al catálogo de M&E como un sistema externo que puede ser reutilizado por otras herramientas.

Para implementar el catálogo de M&E para la herramienta C-INCAMI^{PM} se utilizó un Sistema de Administración de Bases de Datos *MySQL* para el repositorio RDF (*Sesame* también da soporte a *PostgreSQL* y *Oracle*). Se especificó además un esquema RDF que define todos los conceptos y relaciones (metadatos) del marco C-INCAMI que pueden ser reutilizados en los diferentes proyectos de M&E de la organización. Luego, siguiendo este esquema se describieron los datos correspondientes a todas las especificaciones reusables que se utilizan en las actividades de diseño de M&E (atributos, conceptos, modelos de concepto, métricas, etc.). En la Figura 7.23 se transcribe parte de la especificación del Esquema RDF de los conceptos y relaciones del marco C-INCAMI (las definiciones completas del esquema se presentan en el Apéndice A). Luego, en la Figura 7.24 se muestran las descripciones de datos RDF de algunas especificaciones reusables de M&E.

Como se comentó en la Sección 7.2.1.3, este repositorio es accedido por el *Módulo de Acceso al Catálogo C-INCAMI* para ofrecer al subsistema C-INCAMI un servicio transparente de reuso de especificaciones de M&E.

7.2.3. Repositorio del Dominio de Aplicación

Este repositorio corresponde a las bases de datos y otro tipo de sistemas de almacenamiento persistente, propio de la organización, que mantenga información (metadatos y datos) de las entidades y propiedades relevantes a la lógica de negocios de su dominio de aplicación. Tradicionalmente, estos repositorios se implementan en sistemas de gestión de bases de datos relacionales, en los cuales el vocabulario del dominio se encuentra estructurado en tablas interrelacionadas para las cuales es posible definir una conversión al modelo de datos de RDF, tal como se evidencia por los avances en las líneas de investigación citadas en la Sección 7.2.1.3. Brevemente, y sin entrar en demasiados detalles, es posible ver que esta equivalencia es factible. Así, cada tabla de una base de datos relacional puede corresponder, dependiendo de su definición, a una clase RDF (*rdfs:Class*) o propiedad RDF (*rdf:Property*). Las columnas de una tabla correspondiente a una clase equivaldrían a propiedades cuyos valores pueden ser recursos de otras clases (si mantienen claves foráneas) o simplemente literales (*rdfs:Literal*). Las tablas que correspondan a propiedades RDF serán aquellas que mantengan claves foráneas de más de una tabla, estableciendo una relación del tipo muchos-a-muchos.

Como se mencionó anteriormente, la implementación de un repositorio de dominio no fue llevada a cabo en el prototipo desarrollado. No obstante, se ha puesto en claro que su incorporación a la arquitectura propuesta en la Figura 7.1 es totalmente factible.

7.3. Comentarios finales

En este capítulo se ha presentado el diseño de una arquitectura de software que da soporte a la sistematización de proyectos de M&E siguiendo los conceptos del marco C-INCAMI. La arquitectura consta de una aplicación web que puede ser utilizada como herramienta para documentar la información asociada a la gestión de proyectos de M&E. Esta herramienta permite especificar los metadatos

```

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
<rdfs:Class rdf:ID="EntityCategory">
  <rdfs:label xml:lang="en">Entity</rdfs:label>
  <rdfs:comment>Category or type of object which has relevent attributes that can be measured
    for a particular entity of that type.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#RepositoryObject"/>
</rdfs:Class>
...
<rdfs:Class rdf:ID="Attribute">
  <rdfs:label xml:lang="en">Attribute</rdfs:label>
  <rdfs:comment>A measurable physical or abstract property of an entity
category.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#RepositoryObject"/>
  <rdfs:subClassOf rdf:resource="#ContextualEntity"/>
</rdfs:Class>
...
<rdf:Property rdf:ID="Associated_with">
  <rdfs:label xml:lang="en">Associated with</rdfs:label>
  <rdfs:comment>Entity Category to which an attribute is associated.</rdfs:comment>
  <rdfs:domain rdf:resource="#Attribute"/>
  <rdfs:range rdf:resource="#EntityCategory"/>
</rdf:Property>
...
<rdfs:Class rdf:ID="Context">
  <rdfs:label>Context</rdfs:label>
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  <rdfs:comment> ... </rdfs:comment>
</rdfs:Class>
<rdf:Property rdf:ID="ContextDescribedBy">
  <rdfs:label xml:lang="en">Described By</rdfs:label>
  <rdfs:domain rdf:resource="#Context"/>
  <rdfs:range rdf:resource="ContextProperty"/>
</rdf:Property>
...
<rdfs:Class rdf:ID="ContextualEntity">
  <rdfs:label>Contextual Entity</rdfs:label>
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  <rdfs:comment> ... </rdfs:comment>
</rdfs:Class>
<rdf:Property rdf:ID="ContextualEntityApplicableTo">
  <rdfs:label xml:lang="en">Applicable To</rdfs:label>
  <rdfs:domain rdf:resource="#ContextualEntity"/>
  <rdfs:range rdf:resource="Context"/>
</rdf:Property>
...
<rdfs:Class rdf:ID="Metric">
  <rdfs:label xml:lang="en">Metric</rdfs:label>
  <rdfs:comment>The defined measurement or calculation method and the measurement
scale.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#RepositoryObject"/>
  <rdfs:subClassOf rdf:resource="#ContextualEntity"/>
</rdfs:Class>
...
</rdf:RDF>

```

Figura 7.23: Extracto del vocabulario del marco C-INCAMI especificado en RDFS.

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rdf:RDF SYSTEM "http://gidisw.ing.unlpam.edu.ar:8080/entidades.dtd">
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:mae="http://gidisw.ing.unlpam.edu.ar:8080/c_incami.rdf#">
  ...
  <rdf:Description rdf:ID="Web_Application">
    <rdf:type rdf:resource="http://gidisw.ing.unlpam.edu.ar:8080/c_incami.rdf#EntityCategory"/>
    <mae:EntityCategoryName>Web Application</mae:EntityCategoryName>
    <mae:Supercategory>Product</mae:Supercategory>
    <mae:Version>1.0</mae:Version>
  </rdf:Description>
  ...
  <rdf:Description rdf:ID="TargetMarketScope">
    <rdf:type rdf:resource="http://gidisw.ing.unlpam.edu.ar:8080/c_incami.rdf#ContextProperty"/>
    <mae:AttributeName>Target market scope</mae:AttributeName>
    <mae:AttributeDefinition>The geographic scope or range of the market to which
      the web application is aimed .</mae:AttributeDefinition>
    <mae:Associated_with rdf:resource="#Web_Application"/>
    <mae:RelatedContextProperty rdf:resource="#WebsiteOwnersCountry"/>
    <mae:RelatedContextProperty rdf:resource="#TargetGeographicRegion"/>
    <mae:RelatedContextProperty rdf:resource="#SupportedNaturalLanguage"/>
    <mae:Version>1.0</mae:Version>
  </rdf:Description>
  ...
  <rdf:Description rdf:ID="ADN">
    <rdf:type rdf:resource="http://gidisw.ing.unlpam.edu.ar:8080/c_incami.rdf#DirectMetric"/>
    <mae:MetricName>Application Domain Name</mae:MetricName>
    <mae:ValueInterpretation>A categorical symbol which distinguishes the business area
      covered by a software application.</mae:ValueInterpretation>
    <mae:MetricObjective>To characterize the actual context in which an M&E Project
      is executed.</mae:MetricObjective>
    <mae:MetricAccuracy>100%</mae:MetricAccuracy>
    <mae:Quantifies rdf:resource="#ApplicationDomain"/>
    <mae:MetricContains rdf:resource="#ADN_Scale"/>
    <mae:MeasuredUsing rdf:resource="#RQL_Method"/>
    <mae:Version>1.0</mae:Version>
  </rdf:Description>
  <rdf:Description rdf:ID="ADN_Scale">
    <rdf:type rdf:resource="http://gidisw.ing.unlpam.edu.ar:8080/c_incami.rdf#CategoricalScale"/>
    <mae:ScaleType rdf:resource="http://gidisw.ing.unlpam.edu.ar:8080/c_incami.rdf#NOMINAL"/>
    <mae:ValueType rdf:resource="http://gidisw.ing.unlpam.edu.ar:8080/c_incami.rdf#SYMBOL"/>
    <mae:AllowedValue rdf:resource="#E_MAIL"/>
    <mae:AllowedValue rdf:resource="#E_COMMERCE"/>
    <mae:AllowedValue rdf:resource="#E_GOVERNMENT"/>
    <mae:AllowedValue rdf:resource="#E_BANKING"/>
    <mae:AllowedValue rdf:resource="#ADVERTISEMENT"/>
  </rdf:Description>
  <rdf:Description rdf:ID="E_MAIL">
    <rdf:type rdf:resource="http://gidisw.ing.unlpam.edu.ar:8080/c_incami.rdf#Value"/>
    <mae:ValueName>e-mail</mae:ValueName>
    <mae:ValueDescription>...</mae:ValueDescription>
    <mae:Version>1.0</mae:Version>
  </rdf:Description>
  ...
</rdf:RDF>

```

Figura 7.24: Algunas descripciones RDF correspondientes a las especificaciones reusables almacenadas en el Catálogo de M&E.

utilizados en el diseño de la medición y evaluación de entidades relevantes del dominio de aplicación de la organización, incluyendo:

- la especificación de la necesidad de información,
- el contexto en el que se lleva a cabo el proyecto,
- los requerimientos no funcionales a medir y evaluar,
- las métricas que definen cómo se cuantifican dichos requerimientos, y
- los indicadores que definen cómo interpretar las mediciones como un todo, de manera unificada e independientemente de valores de dominio específicos.

La arquitectura también propone un catálogo organizacional semántico de M&E, estructurado también siguiendo las definiciones del marco C-INCAMI, que almacena metadatos reusables en la especificación de requerimientos no funcionales, medición y evaluación, disponible a todos los proyectos de M&E. Este catálogo provee los medios para asegurar que los resultados de M&E sean consistentes entre los diferentes proyectos de la organización, para un mismo período de tiempo así como históricamente.

Adicionalmente, la arquitectura contempla la integración de las especificaciones de M&E, tanto del catálogo como de los proyectos de M&E, con los repositorios de información del dominio de aplicación de la organización de forma recíproca. Así, es posible facilitar la integración de la arquitectura propuesta a la operatoria de la organización, a la vez que se promueve en la misma la incorporación de tecnologías que propician mejoras en los mecanismos disponibles para la manipulación de información, tales como los que proveen las tecnologías propuestas por la Web Semántica.

Finalmente, los dos primeros componentes fueron implementados en un prototipo –según se mostró a lo largo del capítulo–, mostrando la factibilidad de utilizar el marco C-INCAMI propuesto como base para el soporte consistente y coherente de proyectos de medición y evaluación de requerimientos no funcionales en una organización.

Conclusiones y Trabajos Futuros



En esta tesis se ha presentado una propuesta para dar un soporte robusto a la implementación de procesos de medición y evaluación de calidad –o en general, de requerimientos no funcionales-, siguiendo un enfoque orientado a objetivos y sensible al contexto. La propuesta permite integrar dicho soporte a los proyectos de software y web de una organización ofreciendo una herramienta que permite evaluar las entidades involucradas, de interés para la Ingeniería de Software –tales como recursos, productos, procesos, proyectos, entre otros- y en última instancia promover la mejora de los mismos. Aunque la investigación desarrollada en esta tesis se enfoca al dominio de la Ingeniería de Software y Web, la propuesta puede ser aplicada igualmente a cualquier dominio en el que sea posible cuantificar las propiedades de las entidades involucradas.

Cabe destacar que en el desarrollo de esta investigación se han integrado diferentes áreas de conocimiento, a saber: *medición y evaluación*, *computación sensible al contexto*, *web semántica* y *ontologías*. Al respecto, y como se destacará más adelante en este capítulo, la aplicación del enfoque sensible al contexto al área de medición y evaluación conforma una de las principales contribuciones de esta investigación.

A continuación se realiza una revisión de los elementos que conforman la propuesta presentada en esta tesis –destacando los problemas encontrados en el área de medición y evaluación que buscan solucionar (enunciados en el Capítulo 1)- así como las características relevantes de la propuesta. Luego se analizan las contribuciones realizadas a las respectivas áreas de la ciencia. Finalmente se enumeran las líneas de investigación futuras que serán desarrolladas a partir de los resultados presentados en esta tesis.

8.1. Revisión de la Propuesta

8.1.1. Elementos de la Propuesta y Problemas Atacados

El soporte a los procesos de medición y evaluación de una organización propuestos en esta tesis se encuentra conformado por los siguientes elementos o componentes:

- Un *marco conceptual* que define los términos, sus propiedades o atributos y las relaciones que permiten especificar de forma consistente los metadatos y datos utilizados en el diseño e implementación de las actividades de medición y evaluación de requerimientos no funcionales de las entidades involucradas en los proyectos de software y web de una organización (presentado en

los Capítulos 4 y 5). Este marco conceptual, denominado *C-INCAMI* (por *Contextual Information Need, Concept model, Attribute, Metric and Indicator* –conceptos clave del marco) sigue un enfoque orientado a objetivos [Briand *et al.*, 2002; Basili *et al.*, 1994; Park *et al.*, 1996; van Solingen & Berghout, 1997] y sensible al contexto [Dey & Abowd, 1999; Dey, 2001; Brézillon, 2002; Dourish, 2004; Maalej *et al.*, 2006] y además posee una base conceptual ontológica [Martín & Olsina, 2003; Olsina & Martín, 2004] que define formalmente los términos y propiedades utilizadas. Las definiciones del marco C-INCAMI permiten integrar los datos y metadatos del dominio de aplicación –disponibles en los repositorios de información de la organización– a las especificaciones del diseño e implementación de proyectos de medición y evaluación.

- Un *mecanismo de recomendación sensible al contexto* utilizado en el diseño de medición y evaluación así como en la comparación de resultados de diferentes proyectos para determinar su compatibilidad contextual. El mecanismo diseñado se basa en la comparación de descripciones de contexto, utilizando para ello un conjunto de métricas claramente especificadas que definen cómo obtener un valor de similitud entre dos descripciones de contexto para diferentes situaciones (tal como fueron descritos en el Capítulo 5).
- Una *arquitectura de software* que da soporte al marco C-INCAMI y al mecanismo de recomendación sensible al contexto propuestos. Dicha arquitectura provee una herramienta para gestionar proyectos de medición y evaluación de forma consistente a la vez que permite su integración a los sistemas de software de la organización. La arquitectura propuesta incluye entre sus principales componentes:
 - Un *catálogo semántico de M&E* que almacena especificaciones (metadatos) de aquellos elementos de información que pueden ser reutilizados en la especificación de la necesidad de información y el diseño e implementación de la medición y la evaluación de entidades. La implementación del catálogo sigue los lineamientos propuestos por la web semántica. De esta forma, permite almacenar las definiciones de los términos y relaciones utilizados en la especificación de M&E siguiendo la estructura de la ontología en la cual se basa el marco *C-INCAMI*, así como procesar consultas con capacidades de inferencia semántica (tal como se describió en la Sección 7.2.2)
 - Un repositorio (o varios) que almacena los metadatos y datos del dominio correspondiente a la lógica de negocios de la organización, el cual es integrado al marco C-INCAMI utilizando el mecanismo descrito en la Sección 4.6.
 - Una *aplicación web*, llamada *C-INCAMI^{PM}*, en la que se implementan:
 - ▷ los *módulos del marco C-INCAMI*, que definen (i) la estructura de los conceptos, sus atributos y relaciones, permitiendo manipular metadatos y datos asociados al diseño e implementación de M&E y (ii) la lógica de comportamiento asociada a cada concepto (por ejemplo, aquella asociada a la realización de las relaciones entre los conceptos del marco así como el cálculo de métricas indirectas e indicadores).
 - ▷ un *módulo que implementa la lógica de navegación* de la aplicación web de acuerdo al *workflow* del proceso de medición y evaluación adoptado (presentado en la Sección 3.2.2), manipulando las estructuras de datos definidas en los módulos del marco C-INCAMI.
 - ▷ una *memoria de proyectos de M&E* que almacena los metadatos y datos resultantes de la especificación de la necesidad de información, diseño e implementación de medición y evaluación utilizando un formato estructurado que posibilita el intercambio y reuso de información (XML).

- ▷ Un conjunto de módulos responsables de implementar la lógica de acceso, tanto a la memoria de proyectos de M&E como al catálogo semántico de M&E.

El sistema de recomendación sensible al contexto, basado en la comparación de descripciones de contexto, es implementado en la herramienta siguiendo la lógica del *workflow* del proceso de M&E, incorporando las métricas correspondientes (presentadas en la Sección 5.2.1) al catálogo de M&E y utilizándolas para determinar la similitud de las diferentes descripciones de contexto realizadas con el marco C-INCAMI.

El diseño e incorporación de estos elementos a la propuesta permite atacar los problemas encontrados en las áreas relacionadas (tal como fueron enunciados en el Capítulo 1). A continuación se analiza, para cada uno de los problemas asociados a los modelos para medición y evaluación existentes en la literatura, cómo los elementos de esta propuesta proveen una solución:

- **Carencia de una definición clara de los conceptos y relaciones y de una base conceptual sólida:** el marco C-INCAMI provee una definición explícita para cada uno de los conceptos y relaciones incluidos; también se especifican y definen de forma explícita los atributos o propiedades que describen cada uno de los conceptos del marco. La estructura resultante de la especificación de estos conceptos, atributos y relaciones es representada además mediante el lenguaje de modelado UML, en un diagrama de clases, cuya semántica se encuentra explícitamente definida¹. Adicionalmente, la especificación del marco C-INCAMI se basa en una ontología que define los términos, propiedades y relaciones en el dominio de la medición y evaluación de calidad de software [Martín & Olsina, 2003; Olsina & Martín, 2004]. Dicha conceptualización establece un marco de referencia en la interpretación y comunicación de las especificaciones y resultados de medición y evaluación realizadas con el marco C-INCAMI.
- **Incompletitud e inconsistencia entre diferentes propuestas:** el marco C-INCAMI fue diseñado a partir de diferentes fuentes de la literatura relacionada [Briand *et al.*, 2002; Dujmovic, 1996; ISO/IEC 9126-1, 2001; ISO/IEC 15504, 2004; ISO/IEC 14598-1, 1999; ISO/IEC 15939, 2001; Kitchenham *et al.*, 1995; Kan, 2002; Lawler & Kitchenham, 2003; Zuse, 1998], buscando unificar e integrar dichas propuestas en un solo vocabulario. Los términos y relaciones incluidos en el modelo permiten especificar de forma estructurada todos los aspectos requeridos de la medición y evaluación orientada a objetivos: una necesidad de información que describe el objetivo y los requerimientos no funcionales a evaluar, métricas que definen cómo cuantificar los requerimientos no funcionales de bajo nivel e indicadores que definen cómo interpretar los factores de calidad de alto nivel a partir de un conjunto de métricas. De esta forma, el marco C-INCAMI cubre todos los aspectos necesarios y permite la implementación de una solución integral para programas de medición y evaluación.
- **Especificación estructurada de información de contexto inexistente:** respondiendo a la necesidad expresada en trabajos relacionados al área de M&E de conocer y especificar de forma explícita el contexto de las entidades evaluadas (tal como fue presentado en la Sección 2.1.6), el marco C-INCAMI incorpora los términos, atributos y relaciones (descritos en el Capítulo 4 y 5) que permiten especificar de forma estructurada la información de contexto necesaria para establecer un marco coherente en el diseño e interpretación de los resultados de evaluaciones. El modelo conceptual de información de contexto diseñado e incluido en el marco C-INCAMI permite registrar y utilizar esta información en el diseño e interpretación de proyectos de medición y evaluación (como fue mostrado en el Capítulo 6).

¹UML Specification Version 2.4.1 <http://www.omg.org/spec/UML/2.4.1/>

- **Ausencia de un enfoque o mecanismo de integración a la organización:** la estructura del marco conceptual C-INCAMI permite integrar los datos y metadatos del espacio de información del dominio de negocios de la organización a las especificaciones asociadas a las actividades de M&E realizadas con dicho marco (como se describió en la Sección 4.6). Dicha estructura provee, junto a la arquitectura que da soporte al marco (presentada en el Capítulo 7), un mecanismo de integración de la infraestructura de M&E propuesta en esta tesis a la infraestructura de procesos y proyectos existentes en la organización y así facilitar a la misma la implementación, incorporación y el mantenimiento de programas de medición. De forma complementaria, la organización puede recurrir a las guías y recomendaciones, existentes en la literatura relacionada, que cubren los aspectos procedimentales acerca de cómo incorporar las actividades de medición y evaluación a los proyectos de la organización.

También respecto de la aplicación de un *enfoque sensible al contexto*, ya sea independiente del dominio o para el dominio de la Ingeniería de Software, la propuesta presentada en esta tesis presenta una serie de características deseables, algunas de las cuales se encuentran ausentes en las diferentes propuestas encontradas en trabajos relacionados, a saber:

- El modelo de contexto incorporado en el marco C-INCAMI es independiente del dominio ya que se define en función de *propiedades de contexto* y *entidades contextuales* (tal como se describe en las secciones 4.3.1 y 5.1) que pueden ser enlazadas a las definiciones correspondientes del dominio deseado.
- La información de contexto puede ser validada semánticamente contra un modelo de dominio, particularmente el correspondiente al dominio de negocios de la organización. Esto es posible ya que el marco C-INCAMI permite enlazar las especificaciones de las propiedades de contexto (entre otras, tales como categorías de entidad, entidades y atributos) a las definiciones de las propiedades o atributos de las entidades del dominio de la organización especificadas en sus repositorios de información (idealmente siguiendo los lineamientos de la web semántica), tal como se describió en la Sección 4.6.
- Se dispone de un mecanismo de cuantificación robusto para las propiedades de contexto, ya que se basa en el uso de métricas que especifican de forma explícita y estructurada el método y forma de representación de los valores correspondiente (tal como se describe en la Sección 5.1). Así, los valores obtenidos para las propiedades que describen un contexto pueden ser validados y comparados consistentemente.
- El modelo utilizado para describir contexto en el marco C-INCAMI es relativamente simple ya que, una descripción de contexto se conforma de un conjunto de propiedades de contexto y las entidades involucradas de las que se obtienen. Por otro lado, la cuantificación de cada propiedad de contexto no es tan simple como un único valor asignado directamente a la misma (tal como se observa en otras propuestas) sino que se registran como mediciones (y las medidas resultantes) asociadas a las entidades del contexto a las que pertenecen tales propiedades. No obstante, esta complejidad es compensada por el beneficio de contar con un enfoque sólido para la cuantificación de propiedades de contexto, como se mencionó en el punto anterior. Más aún considerando que los mecanismos existentes de especificación y cuantificación fueron reutilizados del marco INCAMI, sobre el cual se construyó la propuesta.
- El modelo de contexto diseñado ha sido integrado en el marco C-INCAMI con un mínimo esfuerzo, ya que reutiliza los términos, atributos y relaciones existentes. Por otro lado, el modelo de contexto del marco C-INCAMI puede ser integrado a los sistemas de una organización con la misma simplicidad. En este caso, las definiciones correspondientes del dominio de negocios de la

organización son utilizadas y enlazadas para especificar las propiedades de contexto relevantes en el marco C-INCAMI. Así, los contextos en los que se llevan a cabo las actividades de medición y evaluación de las entidades de interés para la organización serán descriptos utilizando las propiedades relevantes de su dominio particular.

En resumen, todos los elementos incluidos en esta propuesta –el marco C-INCAMI y la arquitectura asociada– permitirían a una organización implementar programas de medición y evaluación robustos, reduciendo el esfuerzo de puesta en marcha asociado a aspectos técnicos y enfocándose en la definición e implementación de las actividades de medición y evaluación de las entidades de interés de sus proyectos de software y web en pos de alcanzar sus objetivos.

8.1.2. Características de la Propuesta

El marco conceptual C-INCAMI fue diseñado para satisfacer un conjunto de características deseables para proveer una capacidad de medición y evaluación de calidad objetiva, robusta y coherente a los proyectos de software y web de una organización. Tales características, logradas en dicho marco, son descriptas a continuación:

Orientado a objetivos: El marco de medición y evaluación propuesto sigue un enfoque orientado a objetivos (que responde al paradigma *top-down*) en el que todas las tareas de medición y evaluación se enfocan en satisfacer uno o más objetivos, enunciados y especificados de forma clara y explícita al comienzo del proceso, tal como se describe y aplica en trabajos como [Basili *et al.*, 1994; Briand *et al.*, 2002; Card, 2000; Florac *et al.*, 1997; Gresse *et al.*, 1995; IEEE Std. 1061, 2004; ISO/IEC 15939, 2001; Park *et al.*, 1996; van Solingen & Berghout, 1997, 1999]. En el marco C-INCAMI, los objetivos son especificados de forma estructurada mediante una *necesidad de información (InformationNeed)* [Card, 2000; ISO/IEC 15939, 2001; Bailey *et al.*, 2003] en la que se distinguen (tal como se describió en la Sección 4.3 y se ilustró en la Sección 6.2):

- el propósito de la evaluación,
- la característica de calidad (foco) afectada por el propósito,
- la categoría de las entidades que serán objeto de la medición y evaluación,
- el punto de vista del interesado² desde el cual se diseñan las actividades y se interpretan los resultados, y
- una descripción estructurada del contexto relevante a la entidad evaluada y al mismo proceso de medición y evaluación.

Esta especificación de necesidad de información se asocia al proyecto de M&E principal (*MEProject*, tal como se describió en la Figura 4.3), a partir del cual también se definen los subproyectos de medición y evaluación (*MeasurementProject* y *EvaluationProject*, presentados en la Figura 4.2). Así, a partir de esta especificación de necesidad de información se establecen los requerimientos no funcionales –en un modelo de concepto (*ConceptModel*, tal como se presentó también en la Figura 4.3)– para la característica de calidad, foco de la evaluación, en función de características de más bajo nivel y atributos cuantificables de la entidad evaluada. Estos requerimientos no funcionales son usados luego como punto de partida, junto a la especificación de la necesidad de información como referencia, para definir las actividades de medición –durante la selección de métricas para cuantificar los atributos de los requerimientos no funcionales– y evaluación –durante la selección de modelos elementales y globales y criterios de decisión.

²Correspondiente al término inglés *stakeholder*.

Basado en metadatos: El marco C-INCAMI permite especificar no sólo los datos que surgen de los resultados de medición y evaluación (mediante los conceptos *Measurement* y *Evaluation*) sino también los metadatos que establecen cómo se llevarán a cabo las actividades de medición y evaluación (mediante los conceptos *Metric* e *Indicator*, entre otros). Estos metadatos permiten registrar definiciones declarativas y operativas claramente estructuradas para establecer de forma explícita, por un lado, qué será medido y evaluado, y por otro lado, cómo se debe llevar a cabo la medición y evaluación efectiva y objetiva correspondiente [Florac *et al.*, 1997]. Disponer de estos metadatos hace posible que las definiciones correspondientes puedan ser efectivamente comunicadas entre los diferentes agentes interesados e involucrados facilitando así la repetitividad de las mediciones y análisis así como la interpretación de los respectivos resultados. En resumen, los metadatos incluidos en el marco C-INCAMI favorecen a que las especificaciones de medición y evaluación, así como los resultados obtenidos, sean más consistentes, coherentes y confiables.

Centrado en la organización: El marco C-INCAMI provee una base conceptual, no solo para la especificación de las actividades de medición y evaluación, sino también para el soporte y gestión de los procesos de medición y evaluación en proyectos concretos de las organización. Para ello el marco provee la definición de los metadatos correspondientes a los diferentes tipos de proyectos que soportan las actividades de M&E, a los cuales se enlazan los metadatos correspondientes a la especificación de la necesidad de información, los requerimientos no funcionales y del diseño e implementación de las actividades de medición y evaluación (como se describió en la Sección 4.2). Además el marco incluye en la definición de los metadatos los elementos que permiten integrar el espacio de información de dominio de la organización a las especificaciones de M&E (como se describió en la Sección 4.6) que, junto a la arquitectura propuesta (Capítulo 7), facilitarían a la organización la instauración de repositorios de metadatos y datos de medición y evaluación, útiles para establecer líneas base de hechos históricos para análisis y mejoras futuras.

Sensible al contexto: El marco C-INCAMI incluye la definición de los metadatos necesarios para describir de forma clara y estructurada los metadatos y datos necesarios para describir la información del contexto relevante para (i) caracterizar el contexto real de la entidad bajo análisis y (ii) describir el contexto ideal en el que diferentes especificaciones de medición y evaluación sensibles al contexto son aplicables (como se describió en las Secciones 4.3.1 y 5.1). El modelo de contexto diseñado se construye sobre los conceptos y relaciones del mismo marco C-INCAMI utilizados para especificar entidades y sus atributos. De esta forma, es posible utilizar los mecanismos de cuantificación del marco C-INCAMI para describir contextos de la misma forma que son utilizados para cuantificar atributos de entidades con fines de medición, tal como fue propuesto en [Florac *et al.*, 1997]. De la misma forma, el modelo de contexto aprovecha el mecanismo de integración del marco C-INCAMI al dominio de aplicación para obtener descripciones de contexto a partir de los repositorios de información del dominio de la organización, donde se encuentran las descripciones de los procesos, proyectos, recursos, agentes, etc. – que pudieran estar involucradas en el contexto relevante- especificados por la misma para su operatoria. La propuesta también incluye, de forma asociada al modelo de contexto diseñado, los mecanismos necesarios para utilizar la información de contexto en la recomendación de aspectos de diseño y análisis de los resultados, en base a las descripciones de contexto disponibles. Estos mecanismos consisten en un conjunto de métricas que son implementadas sobre las descripciones de contexto correspondientes (según se describió en la Sección 5.2.1).

8.2. Contribuciones

La propuesta presentada en esta tesis constituye un conjunto de contribuciones, en primer lugar, al área de medición y evaluación de calidad en Ingeniería de Software, y en segundo lugar, al área de

*context-awareness*³, destacando además que los resultados obtenidos respecto de esta última fueron aplicados en la primera. A continuación se enumeran los elementos de la propuesta que conforman estas contribuciones:

- Se ha extendido el marco de M&E INCAMI [Olsina *et al.*, 2005, 2008b], sobre el cual se construye la propuesta de esta tesis, incorporando los metadatos que permiten integrar la información del dominio de aplicación de una organización a las especificaciones realizadas con conceptos y relaciones de dicho marco. Estos metadatos permiten aplicar los mecanismos de integración/referenciación de dominios propuestos por la iniciativa de la web semántica⁴.
- Como parte de la instanciación del *enfoque sensible al contexto* [Dey & Abowd, 1999; Dey, 2001; Dourish, 2004; Maalej *et al.*, 2006], se ha definido un modelo de contexto independiente del dominio, utilizando como base conceptual los términos del marco INCAMI que permiten describir una entidad y sus atributos (como fue descrito en la Sección 4.3.1 y 5.1); de esta forma la información de contexto se especifica como un conjunto de propiedades de contexto (*ContextProperty*) –y las entidades (*Entity*) a las cuales se asocian– que pueden ser cuantificadas utilizando métricas (*Metric*) y mediciones (*Measurement*) definidas apropiadamente. Así, la información de contexto puede ser descrita de forma más robusta, tanto semántica como sintácticamente. Este modelo de contexto puede ser adaptado a cualquier dominio de aplicación gracias al mecanismo de integración al dominio de aplicación mencionado en el punto anterior; de esta forma se cubre el mecanismo de captura de información de contexto requerido por el modelo de proceso del enfoque sensible al contexto. El modelo de contexto resultante aprovecha los beneficios de contar con una base conceptual ontológica, tanto para los mismos conceptos que componen el modelo como para las descripciones de contexto de dominios específicos realizadas con los mismos. Adicionalmente, para completar el modelo de proceso del enfoque sensible al contexto, se ha diseñado un mecanismo de procesamiento de información de contexto basado en el uso de un conjunto de métricas que permiten realizar comparaciones de contextos y que conforman el núcleo de un sistema de recomendación sensible al contexto.
- Se aplicó al marco INCAMI el enfoque sensible al contexto mencionado en el punto anterior. De esta forma, ahora es posible describir de forma clara, explícita y estructurada la información del contexto relevante a las actividades de medición y evaluación de calidad de entidades, tal como lo proponen [ISO/IEC 15939, 2001; Park *et al.*, 1996; Basili *et al.*, 1994; Gresse *et al.*, 1995; Briand *et al.*, 2002], entre otros –argumentos que fueron elaborados en la Sección 2.1.6. Utilizando el mecanismo de comparación de contexto diseñado, las descripciones de contexto de M&E son utilizadas en la recomendación de especificaciones reusables en el diseño de medición y evaluación (tales como modelos de concepto, atributos, métricas, modelos de evaluación, entre otros) así como en la interpretación y comparación de resultados de diferentes proyectos. Como resultado de la aplicación de dicho enfoque hemos obtenido un nuevo y mejorado marco conceptual de M&E –principal componente de la propuesta de esta tesis–, al que hemos llamado C-INCAMI (**C**ontextual **I**nformation **N**eed, **C**oncept model, **A**tttribute, **M**etric and **I**ndicator) y que posee las características expuestas previamente en este capítulo. Cabe destacar que, al momento de escritura de esta tesis, no existen en la literatura relacionada propuestas de marcos conceptuales de M&E que apliquen un enfoque sensible al contexto, es decir, que incorporen un modelo de contexto, y mecanismos de captura y procesamiento asociados, que permitan describir de forma clara, explícita y estructurada la información de contexto relevante para la medición y evaluación de calidad de entidades de interés para la Ingeniería de Software y Web.

³Se utiliza el término en inglés ya que se considera que una traducción al español no sería apropiada para describir dicha área de investigación.

⁴W3C Semantic Web Activity [<http://www.w3.org/2001/sw/>]

- En congruencia con la aplicación del enfoque sensible al contexto en el marco C-INCAMI, se ha extendido la ontología de métricas e indicadores [Martín & Olsina, 2003; Olsina & Martín, 2004] para incluir la definición de los conceptos, atributos y relaciones del modelo de contexto diseñado (Secciones 4.3.1 y 5.1 y Apéndices A y B).
- Se desarrolló una prueba de concepto (Capítulo 6) en la cual se ilustra la utilización de las definiciones de términos y relaciones del marco C-INCAMI para llevar a cabo la especificación de los metadatos y datos involucrados en todas las actividades del proceso de M&E en el cual se basa dicho marco (aquel presentado en la Sección 3.2.2). En la prueba de concepto se muestra, además, cómo las descripciones de contexto asociadas a las especificaciones correspondientes son usadas para ofrecer recomendaciones de diseño de requerimientos, medición y evaluación sensibles al contexto.
- Se diseñó una arquitectura de software (Capítulo 7) para dar soporte al marco de M&E C-INCAMI y a los mecanismos asociados –de integración del dominio de aplicación y recomendaciones sensibles al contexto-. La arquitectura diseñada incluye (i) una herramienta web que permite gestionar proyectos de M&E, asistiendo al evaluador en la especificación de los metadatos y datos asociados a la necesidad de información –que incluye la definición clara y explícita de los objetivos de medición, los requerimientos no funcionales y la descripción del contexto relevante-, medición y evaluación, de forma integrada al dominio de aplicación de la organización y apoyado por un sistema de recomendación sensible al contexto, (ii) un catálogo semántico de especificaciones de términos reusables en el diseño de requerimientos, medición y evaluación, y (iii) las conexiones y dependencias con los repositorios existentes de la organización que almacenan los datos correspondientes al dominio de aplicación de la misma.

En resumen, estos desarrollos constituyen un aporte, en primer lugar, al área de *medición y evaluación de calidad de software y web*, al proveer un marco conceptual que efectivamente permite especificar de forma clara, explícita y estructurada los metadatos y datos asociados al diseño de estas actividades. Además, el marco propuesto permite especificar de la misma forma la información del contexto relevante a la M&E de entidades de interés de la Ingeniería de Software y Web, tal como se propone en la literatura relacionada. En segundo lugar, constituye un aporte al área de *context-awareness*, al proveer un modelo de contexto –y mecanismos asociados de captura y procesamiento- independiente del dominio de aplicación y a la vez adaptable mediante los mecanismos provistos por la iniciativa de la web semántica. El modelo de contexto se construye sobre conceptos claros y estructurados del área de metrología (tomados del mismo marco C-INCAMI), reutilizando las definiciones y mecanismos que posibilitan la especificación y cuantificación robusta de los atributos de las entidades pertinentes. El resultado es un modelo de contexto semánticamente y sintácticamente robusto, adaptable a cualquier dominio⁵.

La propuesta presenta una ventaja para su aplicación en entornos industriales principalmente por contar con una base conceptual explícita y estructurada ontológicamente que soporta y facilita la definición de procesos y métodos de M&E organizacionales. A la vez que es posible aplicar enfoques probados industrialmente tales como *Goal-Question-Metric* o *Balanced Scorecard* para la formulación de objetivos y el descubrimiento de atributos de calidad a ser medidos y evaluados. Así, las actividades de medición y evaluación podrían ser más repetibles y consistentes y sus definiciones (diseño) así como sus resultados pueden ser comunicados efectivamente. Al respecto, cabe destacar que el marco conceptual INCAMI (sobre el cual se construyó la propuesta de esta tesis) y proceso asociado fueron aplicados en un caso práctico en una industria de testing [Lew *et al.*, 2011]. Además se cuenta con una definición explícita y detallada del proceso asociado al marco [Becker *et al.*, 2010, 2012]⁶.

⁵Siempre y cuando el dominio de aplicación se encuentre especificado con tecnologías de la web semántica.

⁶Ésta última referencia fue publicada luego de la finalización de la escritura de esta tesis e incorporada aquí durante la revisión de los evaluadores.

La implementación de esta propuesta requeriría, por parte de la organización, cierta madurez en sus procesos de medición y evaluación e infraestructura relacionada, principalmente aquella utilizada para el almacenamiento y reuso de las definiciones de elementos de información asociados a la definición e implementación de procesos de M&E. Como ejemplo, se pueden considerar en este caso organizaciones que cumplan con el nivel 2 de madurez según el modelo *CMMI for Development* [CMMI Product Team, 2006], el cual involucra un *Área de Proceso para Medición y Análisis*, que incluye entre sus prácticas específicas *establecer objetivos de medición, especificar medidas*⁷, *especificar procedimientos de recolección* y almacenamiento de datos así como procedimientos de análisis e *implementar estos procedimientos*. Estas prácticas o actividades son soportadas por la propuesta presentada en esta tesis.

8.3. Líneas de Investigación Futuras

Continuando en la línea de investigación reflejada en esta tesis se propone como trabajo futuro contemplar la posibilidad de aplicar la propuesta presentada en proyectos reales en entornos organizacionales.

Otra de las líneas de investigación en las que se aplicarán los resultados presentados en esta tesis es en el diseño de una estrategia de medición y evaluación que permita enlazar y articular los objetivos de medición a nivel de proyecto (nivel operativo) a los objetivos de alto nivel de la organización (táctico) y de negocios (estratégico) [Basili *et al.*, 2007]. Este enlace es importante porque ayuda a justificar los esfuerzos de medición de software y permite que los resultados contribuyan a las decisiones de alto nivel. El enfoque de medición y evaluación soportado por el marco C-INCAMI, presentado en esta tesis, satisface necesidades de información a nivel de proyecto, sin considerar el impacto que éstas pueden tener con objetivos de niveles organizacionales superiores. Si las necesidades de información se encuentran alineadas a diferentes niveles y la estrategia de medición y evaluación soporta dicha alineación, las necesidades de información de la dirección podrían guiar el desarrollo de programas de medición en niveles operativos; de forma inversa, las necesidades de información de nivel operativo podrían desencadenar futuras políticas y estrategias organizacionales a nivel gerencial.

La estrategia que se propone diseñar, llamada GOCAME+ (*Goal-Oriented, Context-Aware Measurement and Evaluation Plus*) –a partir de la estrategia GOCAME, citada en el Capítulo 3- proveerá mecanismos para dar soporte a la especificación integrada y enlazada de necesidades de información proveniente de distintos niveles de una organización (operativo, táctico y estratégico). Esta estrategia constará de tres pilares –un marco conceptual de medición y evaluación, un proceso de medición y evaluación; y métodos y herramientas- y permitirá instanciar modelos de calidad para cada nivel, estableciendo las dependencias entre ellos, guiando la implementación de los procesos de M&E que permitan satisfacer las distintas necesidades de información hasta los niveles más altos. Para tal fin se pretende readecuar y modelar, por un lado, el marco conceptual de M&E, presentado en esta tesis, para que contemple múltiples instancias de necesidades de información de distintos niveles organizacionales (operativo, táctico y estratégico), atendiendo a diferentes focos de calidad; y por otro lado, el proceso de M&E para especificar las actividades que se planifican y ejecutan (principalmente en la etapa de definición de requerimientos no funcionales) en los distintos niveles organizacionales para satisfacer las distintas necesidades de información.

En una segunda línea de investigación, se pretende utilizar el marco de M&E C-INCAMI, junto a la estrategia GOCAME+, citada en el párrafo anterior, para dar soporte a procesos de *gestión de riesgos de software* (*Software Risk Management* o *SRM*). SRM es una parte fundamental de los procesos de gestión de proyectos que permite llevar a cabo acciones para prevenir la existencia de riesgos o reducir su impacto en las entidades de interés, en lugar de tratar con sus consecuencias. Para ello, existen un conjunto de políticas, procesos, métodos, técnicas y herramientas que permiten establecer el contexto

⁷Definido en el documento citado como “*Variable to which a value is assigned as a result of measurement*”.

en el que pueden existir los riesgos, así como evaluarlos y comprenderlos, tratarlos –preparando acciones preventivas y perfectivas-, controlarlos y comunicar su estado, evitando de esta forma comprometer los diferentes aspectos de la entidad de interés, tales como calidad, tiempo, costo y capacidades funcionales.

El enfoque de evaluación propuesto por la estrategia GOCAME podría ser utilizado entonces para evaluar y comprender los riesgos, controlando los niveles de satisfacción de calidad de los atributos de las entidades de interés asociados a algún riesgo, como parte de los requerimientos no funcionales. Especificando los atributos, métricas e indicadores apropiados para medir e interpretar el estado de los riesgos, será posible determinar si estos fueron reducidos y en qué porcentaje. A mayores niveles de calidad alcanzados por tales atributos, menor serán los riesgos asociados. Luego de recomendar y aplicar los cambios pertinentes en los atributos de la entidad, las vulnerabilidades pueden ser reevaluadas para determinar si éstas fueron reducidas, así como el riesgo asociado. Estas actividades de evaluación, aplicación de cambios y reevaluación son realizadas de forma iterativa para establecer un proceso continuo de análisis y reducción de riesgos. Usando este enfoque se espera entonces robustecer la evaluación de riesgos y mejorar así el conocimiento organizacional asociado a los mismos.



Conceptualización del Modelo de Contexto

En este apéndice se presentan las definiciones de los conceptos, atributos y relaciones del modelo de contexto diseñado (presentado en las secciones 4.3.1 y 5.1). También se incluyen un conjunto de axiomas que imponen restricciones estructurales a las relaciones entre los conceptos del modelo. Para ello se utiliza el lenguaje OCL (*Object Constraint Language*), diseñado para ser utilizado junto a UML para dicho propósito. La conceptualización presentada a continuación pretende extender la ontología de métricas e indicadores, presentada en [Martín & Olsina, 2003; Olsina & Martín, 2004], para dar soporte ontológico al nuevo marco C-INCAMI presentado en esta tesis.

CONCEPTO Context: Una entidad especial, descrita por un conjunto de propiedades de contexto que caracterizan el estado relevante de la situación de una entidad de interés dada. La situación de una entidad involucra la tarea y propósito para con esa entidad y sus relaciones con otras entidades respecto de dicha tarea y propósito. Un contexto puede ser caracterizado midiendo las propiedades (atributos) de las entidades involucradas en él.

RELACIÓN describedBy: Un contexto está descrito por una o más propiedades de contexto.

RELACIÓN quantifiedBy: Un contexto es cuantificado por una o más métricas (utilizadas para asignar valores a las propiedades de contexto que lo describen).

CONCEPTO ContextProperty: Un atributo que describe el *Contexto* de una entidad dada; este atributo pertenece a (la categoría de) alguna de las entidades que participan en el *Contexto* descrito.

ATRIBUTO weight: Valor que refleja la importancia relativa de la propiedad dentro de la descripción del contexto correspondiente; los posibles valores se encuentran en un rango entre 0,0 y 1,0.

ATRIBUTO relevance: Indica la pertinencia de usar la propiedad para describir el contexto de una entidad particular.

ATRIBUTO multiplicityOfValues: Indica la cantidad de valores (mediciones) que pueden ser asignados a la propiedad de contexto.

RELACIÓN relatedContextProperty: Una propiedad de contexto puede estar relacionada semánticamente a cero o más propiedades de contexto.

RELACIÓN comparedUsing: Una propiedad de contexto es comparada a otra propiedad de contexto equivalente utilizando un criterio de similitud.

CONCEPTO **ContextualEntity:** Una entidad cuyo uso o interpretación correcta es sensible al contexto en el cual será aplicada y/o analizada.

RELACIÓN **applicableTo** Una o más entidades contextuales son aplicables a un contexto destino (*targetContext*).

CONCEPTO **SimilarityCriterion:** Patrón de valoración que permite determinar la similitud semántica entre los valores especificados para una misma propiedad de contexto en diferentes descripciones de contexto.

CONCEPTO **ThresholdSimilarityCriterion:** Criterio de similitud basado en un patrón de umbral para determinar la similitud semántica entre los valores de una propiedad de contexto en diferentes descripciones de contexto.

ATRIBUTO **matchingDirection:** Dirección relativa a un valor umbral hacia la cual los valores encontrados en la escala correspondiente satisfacen el criterio de similitud de umbral.

RELACIÓN **threshold:** Un criterio de similitud de umbral tiene como valor umbral una medición (*Measurement*).

CONCEPTO **RangeSimilarityCriterion:** Criterio de similitud basado en un patrón de rango para determinar la similitud semántica entre los valores de una propiedad de contexto en diferentes descripciones de contexto.

RELACIÓN **upperThreshold:** Un criterio de similitud de rango tiene como valor umbral superior una medición (*Measurement*).

RELACIÓN **lowerThreshold:** Un criterio de similitud de rango tiene como valor umbral inferior una medición (*Measurement*).

CONCEPTO **ExactSimilarityCriterion:** Criterio de similitud basado en un patrón de equivalencias uno a uno agregadas de forma lógica (mediante un agregador lógico) para determinar la similitud semántica entre los valores de una propiedad de contexto en diferentes descripciones de contexto.

RELACIÓN **logicAggregator:** Un criterio de similitud exacto posee un agregador lógico (*LogicAggregator*).

RELACIÓN **valueMatch:** Un criterio de similitud exacto especifica una o más mediciones (*Measurement*) como valores que satisfacen el criterio.

CONCEPTO **LogicAggregator:** Operador lógico utilizado por un criterio de similitud exacto para agregar en un único valor los resultados parciales de cada equivalencia uno a uno.

AXIOMA Las coincidencias de valor (*valueMatch*) de un criterio de similitud exacto (*ExactSimilarityCriterion*) son mediciones (*Measurement*) generadas a partir de la métrica (*Metric*) que cuantifica la propiedad de contexto (*ContextProperty*) a la cual se encuentra asignado dicho criterio.

OCL: Context Metric: Metric.allInstance->forAll(m:Metric|m.quantifies.oclType='ContextProperty' and m.quantifies.comparedUsing.oclType='ExactSimilarityCriterion' implies m.quantifies.comparedUsing.valueMatch->forAll(vm:Measurement|vm.refersTo=m))

AXIOMA El umbral (threshold) asignado a un criterio de similitud de umbral (*ThresholdSimilarityCriterion*) corresponde a una medición (*Measurement*) generada a partir de la métrica (*Metric*) que cuantifica la propiedad de contexto (*ContextProperty*) a la cual se asignó dicho criterio.

OCL: Context Metric: Metric.allInstance->forAll(m:Metric|m.quantifies.oclType='ContextProperty' and m.quantifies.comparedUsing.oclType='ThresholdSimilarityCriterion' implies m.quantifies.comparedUsing.threshold.refersTo=m)

AXIOMA Los umbrales superior (*upperThreshold*) e inferior(*lowerThreshold*) asignados a un criterio de similitud de rango (*RangeSimilarityCriterion*) corresponden a mediciones generadas a partir de la métrica (*Metric*) que cuantifica la propiedad de contexto (*ContextProperty*) a la cual se asignó dicho criterio.

OCL: Context Metric: Metric.allInstance->forAll(m:Metric|m.quantifies.oclType='ContextProperty' and m.quantifies.comparedUsing.oclType='RangeSimilarityCriterion' implies m.quantifies.comparedUsing.upperThreshold.refersTo=m and m.quantifies.comparedUsing.lowerThreshold.refersTo=m)



Esquema RDF del Marco C-INCAMI

En este apéndice se transcribe la especificación en RDF correspondiente a la definición de las clases, atributos y relaciones del marco C-INCAMI, tal como fueron almacenados en el catálogo de M&E descrito en la arquitectura de software que le da soporte (ver Sección 7.2.2). Los atributos son especificados como propiedades RDF, al igual que las relaciones, utilizando como identificador el nombre del atributo anteponiendo el nombre de la clase a la cual pertenecen. Adicionalmente, el rango (*rdfs:range*) de estas propiedades corresponde al recurso *rdfs:Literal* (<http://www.w3.org/2000/01/rdf-schema#Literal>).

Listado A.1: Código RDF correspondiente a la definición de las clases, atributos y relaciones del marco C-INCAMI.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
<!-- =====RepositoryObject y sus propiedades===== -->
<
<rdfs:Class rdf:ID="RepositoryObject">
  <rdfs:label xml:lang="en">Repository Object</rdfs:label>
  <rdfs:comment>Any object in the repository.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
</rdfs:Class>
<rdf:Property rdf:ID="Version">
  <rdfs:label xml:lang="en">Version</rdfs:label>
  <rdfs:domain rdf:resource="#RepositoryObject"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<!-- =====EntityCategory y sus propiedades===== -->
<rdfs:Class rdf:ID="EntityCategory">
  <rdfs:label xml:lang="en">Entity</rdfs:label>
  <rdfs:comment>Category or type of object which has relevent attributes that can be measured
  for a particular entity of that type.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#RepositoryObject"/>
</rdfs:Class>
<rdf:Property rdf:ID="EntityCategoryName">
  <rdfs:label xml:lang="en">Name</rdfs:label>
  <rdfs:domain rdf:resource="#EntityCategory"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
```

```

<rdf:Property rdf:ID="EntityCategoryDescription">
  <rdfs:label xml:lang="en">Description</rdfs:label>
  <rdfs:domain rdf:resource="#EntityCategory"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="Supercategory">
  <rdfs:label xml:lang="en">Supercategory</rdfs:label>
  <rdfs:domain rdf:resource="#EntityCategory"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="RelatedEntityCategory">
  <rdfs:label xml:lang="en">Related Entity Category</rdfs:label>
  <rdfs:domain rdf:resource="#EntityCategory"/>
  <rdfs:range rdf:resource="#EntityCategory"/>
</rdf:Property>
<!-- =====Attribute y sus propiedades===== -->
<rdfs:Class rdf:ID="Attribute">
  <rdfs:label xml:lang="en">Attribute</rdfs:label>
  <rdfs:comment>A measurable physical or abstract property of an entity category.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#RepositoryObject"/>
  <rdfs:subClassOf rdf:resource="#ContextualEntity"/>
</rdfs:Class>
<rdf:Property rdf:ID="AttributeName">
  <rdfs:label xml:lang="en">Name</rdfs:label>
  <rdfs:domain rdf:resource="#Attribute"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="AttributeDefinition">
  <rdfs:label xml:lang="en">Definition</rdfs:label>
  <rdfs:domain rdf:resource="#Attribute"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="AttributeObjetive">
  <rdfs:label xml:lang="en">Objetive</rdfs:label>
  <rdfs:domain rdf:resource="#Attribute"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="Associated_with">
  <rdfs:label xml:lang="en">Associated with</rdfs:label>
  <rdfs:comment>Entity Category to which an attribute is associated.</rdfs:comment>
  <rdfs:domain rdf:resource="#Attribute"/>
  <rdfs:range rdf:resource="#EntityCategory"/>
</rdf:Property>
<!-- =====Context===== -->
<rdfs:Class rdf:ID="Context">
  <rdfs:label>Context</rdfs:label>
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  <rdfs:comment>A special entity, described by a set of context properties characterizing the relevant state of the
  situation of an entity of interest. The entity's situation involves the task and purpose towards that entity and
  its
  relationships with other entities regarding that task and purpose.</rdfs:comment>
</rdfs:Class>
<rdf:Property rdf:ID="ContextDescribedBy">
  <rdfs:label xml:lang="en">Described By</rdfs:label>
  <rdfs:domain rdf:resource="#Context"/>
  <rdfs:range rdf:resource="ContextProperty"/>
</rdf:Property>
<!-- =====ContextualEntity y sus propiedades===== -->
<rdfs:Class rdf:ID="ContextualEntity">

```

```

    <rdfs:label>Contextual Entity</rdfs:label>
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <rdfs:comment>An entity whose correct use or interpretation is sensitive to the context in which
        it will be applied and/or used.</rdfs:comment>
</rdfs:Class>
<rdf:Property rdf:ID="ContextualEntityApplicableTo">
    <rdfs:label xml:lang="en">Applicable To</rdfs:label>
    <rdfs:domain rdf:resource="#ContextualEntity"/>
    <rdfs:range rdf:resource="Context"/>
</rdf:Property>
<!-- =====ContextProperty y sus propiedades===== -->
<rdfs:Class rdf:ID="ContextProperty">
    <rdfs:label>Context Property</rdfs:label>
    <rdfs:subClassOf rdf:resource="#Attribute"/>
    <rdfs:comment>An attribute describing the context of a given entity, this attribute is associated with
        the entity category to which some of the entities involved in the described context belongs.</rdfs:comment>
</rdfs:Class>
<rdf:Property rdf:ID="ContextPropertyWeight">
    <rdfs:label xml:lang="en">Weight</rdfs:label>
    <rdfs:domain rdf:resource="#ContextProperty"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="ContextPropertyRelevance">
    <rdfs:label xml:lang="en">Relevance</rdfs:label>
    <rdfs:domain rdf:resource="#ContextProperty"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="ContextPropertyMultiplicityOfValues">
    <rdfs:label xml:lang="en">Multiplicity of Values</rdfs:label>
    <rdfs:domain rdf:resource="#ContextProperty"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="RelatedContextProperty">
    <rdfs:label xml:lang="en">Related Context Property</rdfs:label>
    <rdfs:domain rdf:resource="#ContextProperty"/>
    <rdfs:range rdf:resource="#ContextProperty"/>
</rdf:Property>
<!-- =====Calculable Concept y sus propiedades===== -->
<rdfs:Class rdf:ID="CalculableConcept">
    <rdfs:label xml:lang="en">Calculable Concept</rdfs:label>
    <rdfs:comment>Abstract relationship between attributes of entities and information needs </rdfs:comment>
    <rdfs:subClassOf rdf:resource="#RepositoryObject"/>
</rdfs:Class>
<rdf:Property rdf:ID="CalculableConceptName">
    <rdfs:label xml:lang="en">Name</rdfs:label>
    <rdfs:domain rdf:resource="#CalculableConcept"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="CalculableConceptDefinition">
    <rdfs:label xml:lang="en">Definition</rdfs:label>
    <rdfs:domain rdf:resource="#CalculableConcept"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="CalculableConceptReference">
    <rdfs:label xml:lang="en">Reference</rdfs:label>
    <rdfs:domain rdf:resource="#CalculableConcept"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="Represented_by">

```

```

    <rdfs:label xml:lang="en">Represented by</rdfs:label>
    <rdfs:comment>A calculable concept can by represented by none or several concept models.</rdfs:comment>
    <rdfs:domain rdf:resource="#CalculableConcept"/>
    <rdfs:range rdf:resource="#ConceptModel"/>
</rdf:Property>
<rdf:Property rdf:ID="Combines">
    <rdfs:label xml:lang="en">Combines</rdfs:label>
    <rdfs:comment>A calculable concept may combine one or more attributes.</rdfs:comment>
    <rdfs:domain rdf:resource="#CalculableConcept"/>
    <rdfs:range rdf:resource="#Attribute"/>
</rdf:Property>
<rdf:Property rdf:ID="Subconcept">
    <rdfs:label xml:lang="en">subconcept</rdfs:label>
    <rdfs:comment>A calculable concept may be described by other calculable concepts.</rdfs:comment>
    <rdfs:domain rdf:resource="#CalculableConcept"/>
    <rdfs:range rdf:resource="#CalculableConcept"/>
</rdf:Property>
<!-- ===== Concept Model y sus propiedades ===== -->
<rdfs:Class rdf:ID="ConceptModel">
    <rdfs:label xml:lang="en">Concept Model</rdfs:label>
    <rdfs:comment> General definition for the concept model wich represents the requirements
        for the associated concept.</rdfs:comment>
    <rdfs:subClassOf rdf:resource="#RepositoryObject"/>
    <rdfs:subClassOf rdf:resource="#ContextualEntity"/>
</rdfs:Class>
<rdf:Property rdf:ID="ConceptModelName">
    <rdfs:label xml:lang="en">Name</rdfs:label>
    <rdfs:domain rdf:resource="#ConceptModel"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="ConceptModelSpecification">
    <rdfs:label xml:lang="en">Specification</rdfs:label>
    <rdfs:domain rdf:resource="#ConceptModel"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="ConceptModelReference">
    <rdfs:label xml:lang="en">Reference</rdfs:label>
    <rdfs:domain rdf:resource="#ConceptModel"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="ConceptModelType">
    <rdfs:label xml:lang="en">Type</rdfs:label>
    <rdfs:comment> ... </rdfs:comment>
    <rdfs:domain rdf:resource="#ConceptModel"/>
    <rdfs:range rdf:resource="#TypeOfConceptModel"/>
</rdf:Property>
<rdfs:Class rdf:ID="TypeOfConceptModel">
    <rdfs:label xml:lang="en">Concept Model Type</rdfs:label>
    <rdfs:comment> ... </rdfs:comment>
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
</rdfs:Class>
<rdf:Description rdf:ID="OWN">
    <rdf:type rdf:resource="#TypeOfConceptModel"/>
    <rdfs:label xml:lang="en">Own</rdfs:label>
</rdf:Description>
<rdf:Description rdf:ID="STANDARD">
    <rdf:type rdf:resource="#TypeOfConceptModel"/>
    <rdfs:label xml:lang="en">Standard</rdfs:label>
</rdf:Description>

```

```

<rdf:Description rdf:ID="MIXTURE">
  <rdf:type rdf:resource="#TypeOfConceptModel"/>
  <rdfs:label xml:lang="en">Mixture</rdfs:label>
</rdf:Description>
<rdf:Property rdf:ID="ConceptModelConstraints">
  <rdfs:label xml:lang="en">Constraints</rdfs:label>
  <rdfs:domain rdf:resource="#ConceptModel"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="Specified_by">
  <rdfs:label xml:lang="en">Specified by</rdfs:label>
  <rdfs:comment>A concept model is specified by a requirement tree.</rdfs:comment>
  <rdfs:domain rdf:resource="#ConceptModel"/>
  <rdfs:range rdf:resource="#RequirementTree"/>
</rdf:Property>
<!-- =====RequirementTree y sus propiedades===== -->
<rdfs:Class rdf:ID="RequirementTree">
  <rdfs:label xml:lang="en">Requirement Tree</rdfs:label>
  <rdfs:comment>Set of concept nodes and attribute nodes defining the relationship between subconcepts and
    attributes that models a concept and provides the foundations for specifying his requirements and further
    evaluation or estimation as a whole.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#RepositoryObject"/>
</rdfs:Class>
<rdf:Property rdf:ID="RequirementTreeRootNode">
  <rdfs:label xml:lang="en">Root Node</rdfs:label>
  <rdfs:domain rdf:resource="#RequirementTree"/>
  <rdfs:range rdf:resource="#CalculableConceptNode"/>
</rdf:Property>
<!-- =====RequirementTreeNode y sus propiedades===== -->
<rdfs:Class rdf:ID="RequirementTreeNode">
  <rdfs:label xml:lang="en">Requirement Tree Node</rdfs:label>
  <rdfs:comment>Abstract representation of a node in a requirement tree.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#RepositoryObject"/>
</rdfs:Class>
<!-- =====CalculableConceptNode y sus propiedades===== -->
<rdfs:Class rdf:ID="CalculableConceptNode">
  <rdfs:label xml:lang="en">Calculable Concept Node</rdfs:label>
  <rdfs:comment>A requirement tree node that represents a concept included in a concept model.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#RequirementTreeNode"/>
</rdfs:Class>
<rdf:Property rdf:ID="CalculableConceptNodeChild">
  <rdfs:label xml:lang="en">Child</rdfs:label>
  <rdfs:comment>A calculable concept node can have several child nodes.</rdfs:comment>
  <rdfs:domain rdf:resource="#CalculableConceptNode"/>
  <rdfs:range rdf:resource="#RequirementTreeNode"/>
</rdf:Property>
<rdf:Property rdf:ID="ReferencedConcept">
  <rdfs:label xml:lang="en">Referenced Concept</rdfs:label>
  <rdfs:comment>A calculable concept node references a calculable concept included in
    a concept model.</rdfs:comment>
  <rdfs:domain rdf:resource="#CalculableConceptNode"/>
  <rdfs:range rdf:resource="#CalculableConcept"/>
</rdf:Property>
<!-- =====AttributeNode y sus propiedades===== -->
<rdfs:Class rdf:ID="AttributeNode">
  <rdfs:label xml:lang="en">Attribute Node</rdfs:label>
  <rdfs:comment>A requirement tree node that represents an attribute included in a concept model.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#RequirementTreeNode"/>
</rdfs:Class>

```

```

<rdf:Property rdf:ID="ReferencedAttribute">
  <rdfs:label xml:lang="en">Referenced Attribute</rdfs:label>
  <rdfs:comment>An attribute node references an attribute included in a concept model.</rdfs:comment>
  <rdfs:domain rdf:resource="#AttributeNode"/>
  <rdfs:range rdf:resource="#Attribute"/>
</rdf:Property>
<!-- =====Metric y sus propiedades===== -->
<rdfs:Class rdf:ID="Metric">
  <rdfs:label xml:lang="en">Metric</rdfs:label>
  <rdfs:comment>The defined measurement or calculation method and the measurement scale.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#RepositoryObject"/>
  <rdfs:subClassOf rdf:resource="#ContextualEntity"/>
</rdfs:Class>
<rdf:Property rdf:ID="MetricName">
  <rdfs:label xml:lang="en">Name</rdfs:label>
  <rdfs:domain rdf:resource="#Metric"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="ValueInterpretation">
  <rdfs:label xml:lang="en">Value Interpretation</rdfs:label>
  <rdfs:domain rdf:resource="#Metric"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="MetricObjective">
  <rdfs:label xml:lang="en">Objective</rdfs:label>
  <rdfs:domain rdf:resource="#Metric"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
  <rdfs:comment>Goal or purpose for applying this metric.</rdfs:comment>
</rdf:Property>
<rdf:Property rdf:ID="MetricReference">
  <rdfs:label xml:lang="en">Reference</rdfs:label>
  <rdfs:domain rdf:resource="#Metric"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="MetricAccuracy">
  <rdfs:label xml:lang="en">Accuracy</rdfs:label>
  <rdfs:domain rdf:resource="#Metric"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="MetricAuthor">
  <rdfs:label xml:lang="en">Author</rdfs:label>
  <rdfs:domain rdf:resource="#Metric"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="Quantifies">
  <rdfs:label xml:lang="en">Quantifies</rdfs:label>
  <rdfs:domain rdf:resource="#Metric"/>
  <rdfs:range rdf:resource="#Attribute"/>
</rdf:Property>
<rdf:Property rdf:ID="MetricContains">
  <rdfs:label xml:lang="en">Contains</rdfs:label>
  <rdfs:domain rdf:resource="#Metric"/>
  <rdfs:range rdf:resource="#Scale"/>
</rdf:Property>
<!-- =====DirectMetric y sus propiedades===== -->
<rdfs:Class rdf:ID="DirectMetric">
  <rdfs:label xml:lang="en">Direct Metric</rdfs:label>
  <rdfs:comment>A metric of an attribute that does not depend upon a metric of any other attribute.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#Metric"/>

```

```

</rdfs:Class>
<rdf:Property rdf:ID="MeasuredUsing">
  <rdfs:label xml:lang="en">Measured Using</rdfs:label>
  <rdfs:domain rdf:resource="#DirectMetric"/>
  <rdfs:range rdf:resource="#MeasurementMethod"/>
</rdf:Property>
<!-- =====IndirectMetric y sus propiedades===== -->
<rdfs:Class rdf:ID="IndirectMetric">
  <rdfs:label xml:lang="en">Indirect Metric</rdfs:label>
  <rdfs:comment>A metric of an attribute that is derived from metrics of one or more other attributes.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#Metric"/>
</rdfs:Class>
<rdf:Property rdf:ID="CalculatedUsing">
  <rdfs:label xml:lang="en">Calculated Using</rdfs:label>
  <rdfs:domain rdf:resource="#IndirectMetric"/>
  <rdfs:range rdf:resource="#CalculationMethod"/>
</rdf:Property>
<rdf:Property rdf:ID="RelatedMetric">
  <rdfs:label xml:lang="en">Related Metric</rdfs:label>
  <rdfs:domain rdf:resource="#IndirectMetric"/>
  <rdfs:range rdf:resource="#Metric"/>
</rdf:Property>
<rdf:Property rdf:ID="SpecifiedBy">
  <rdfs:label xml:lang="en">Specified By</rdfs:label>
  <rdfs:domain rdf:resource="#IndirectMetric"/>
  <rdfs:range rdf:resource="#Function"/>
</rdf:Property>
<!-- =====Scale y sus propiedades===== -->
<rdfs:Class rdf:ID="Scale">
  <rdfs:label xml:lang="en">Scale</rdfs:label>
  <rdfs:comment>A set of values with defined properties.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#RepositoryObject"/>
</rdfs:Class>
<rdf:Property rdf:ID="ScaleType">
  <rdfs:label xml:lang="en">Scale Type</rdfs:label>
  <rdfs:domain rdf:resource="#Scale"/>
  <rdfs:range rdf:resource="#TypeOfScale"/>
</rdf:Property>
<rdf:Property rdf:ID="ValueType">
  <rdfs:label xml:lang="en">Value Type</rdfs:label>
  <rdfs:domain rdf:resource="#Scale"/>
  <rdfs:range rdf:resource="#TypeOfValue"/>
</rdf:Property>
<rdfs:Class rdf:ID="TypeOfScale">
  <rdfs:label xml:lang="en">Type of Scale</rdfs:label>
  <rdfs:comment> ... </rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
</rdfs:Class>
<rdf:Description rdf:ID="NOMINAL">
  <rdf:type rdf:resource="#TypeOfScale"/>
  <rdfs:label xml:lang="en">Nominal</rdfs:label>
</rdf:Description>
<rdf:Description rdf:ID="ORDINAL">
  <rdf:type rdf:resource="#TypeOfScale"/>
  <rdfs:label xml:lang="en">Ordinal</rdfs:label>
</rdf:Description>
<rdf:Description rdf:ID="INTERVAL">
  <rdf:type rdf:resource="#TypeOfScale"/>
  <rdfs:label xml:lang="en">Interval</rdfs:label>

```

```

</rdf:Description>
<rdf:Description rdf:ID="RATIO">
  <rdf:type rdf:resource="#TypeOfScale"/>
  <rdfs:label xml:lang="en">Ratio</rdfs:label>
</rdf:Description>
<rdf:Description rdf:ID="ABSOLUTE">
  <rdf:type rdf:resource="#TypeOfScale"/>
  <rdfs:label xml:lang="en">Absolute</rdfs:label>
</rdf:Description>
<rdfs:Class rdf:ID="TypeOfValue">
  <rdfs:label xml:lang="en">Type of Value</rdfs:label>
  <rdfs:comment> ... </rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
</rdfs:Class>
<rdf:Description rdf:ID="INTEGER">
  <rdf:type rdf:resource="#TypeOfValue"/>
  <rdfs:label xml:lang="en">Integer</rdfs:label>
</rdf:Description>
<rdf:Description rdf:ID="REAL">
  <rdf:type rdf:resource="#TypeOfValue"/>
  <rdfs:label xml:lang="en">Real</rdfs:label>
</rdf:Description>
<rdf:Description rdf:ID="SYMBOL">
  <rdf:type rdf:resource="#TypeOfValue"/>
  <rdfs:label xml:lang="en">Symbol</rdfs:label>
</rdf:Description>
<!-- =====CategoricalScale y sus propiedades===== -->
<rdfs:Class rdf:ID="CategoricalScale">
  <rdfs:label xml:lang="en">Categorical Scale</rdfs:label>
  <rdfs:comment>A scale where the measured or calculated values are categories, and can not be
    expressed in units, in an strict sense.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#Scale"/>
</rdfs:Class>
<rdf:Property rdf:ID="AllowedValue">
  <rdfs:comment>A list of comma-separated values allowed for the categorical scale</rdfs:comment>
  <rdfs:label xml:lang="en">Allowed Value</rdfs:label>
  <rdfs:domain rdf:resource="#CategoricalScale"/>
  <rdfs:range rdf:resource="#CategoricalValue"/>
</rdf:Property>
<rdfs:Class rdf:ID="CategoricalValue">
  <rdfs:label xml:lang="en">Value</rdfs:label>
  <rdfs:subClassOf rdf:resource="#RepositoryObject"/>
  <rdfs:comment>Constant used to describe the value of an attribute or property.</rdfs:comment>
</rdfs:Class>
<rdf:Property rdf:ID="CategoricalValueName">
  <rdfs:comment>Name of a value</rdfs:comment>
  <rdfs:label xml:lang="en">Name</rdfs:label>
  <rdfs:domain rdf:resource="#CategoricalValue"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="CategoricalValueDescription">
  <rdfs:comment>Description of a value</rdfs:comment>
  <rdfs:label xml:lang="en">Description</rdfs:label>
  <rdfs:domain rdf:resource="#CategoricalValue"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<!-- =====NumericalScale y sus propiedades===== -->
<rdfs:Class rdf:ID="NumericalScale">
  <rdfs:label xml:lang="en">NumericalScale</rdfs:label>

```

```

    <rdfs:comment>A scale where the measured or calculated values are numbers that can be expressed in units,
    in an strict sense.</rdfs:comment>
    <rdfs:subClassOf rdf:resource="#Scale"/>
</rdfs:Class>
<rdf:Property rdf:ID="RangeMinValue">
    <rdfs:label xml:lang="en">Numerical Scale Type</rdfs:label>
    <rdfs:domain rdf:resource="#NumericalScale"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="RangeMaxValue">
    <rdfs:label xml:lang="en">Expressed In</rdfs:label>
    <rdfs:domain rdf:resource="#NumericalScale"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="StepValue">
    <rdfs:label xml:lang="en">Expressed In</rdfs:label>
    <rdfs:domain rdf:resource="#NumericalScale"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="Representation">
    <rdfs:label xml:lang="en">Numerical Scale Type</rdfs:label>
    <rdfs:domain rdf:resource="#NumericalScale"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="ExpressedIn">
    <rdfs:label xml:lang="en">Expressed In</rdfs:label>
    <rdfs:domain rdf:resource="#NumericalScale"/>
    <rdfs:range rdf:resource="#Unit"/>
</rdf:Property>
<!-- =====Unit y sus propiedades===== -->
<rdfs:Class rdf:ID="Unit">
    <rdfs:label xml:lang="en">Unit</rdfs:label>
    <rdfs:comment>Particular quantity, defined and adopted by convention, with which other quantities of
    the same kind are compared in order to express their magnitude relative to that quantity </rdfs:comment>
    <rdfs:subClassOf rdf:resource="#RepositoryObject"/>
</rdfs:Class>
<rdf:Property rdf:ID="UnitName">
    <rdfs:label xml:lang="en">Name</rdfs:label>
    <rdfs:domain rdf:resource="#Unit"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="UnitDescription">
    <rdfs:label xml:lang="en">Description</rdfs:label>
    <rdfs:domain rdf:resource="#Unit"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="Acronym">
    <rdfs:label xml:lang="en">Acronym</rdfs:label>
    <rdfs:domain rdf:resource="#Unit"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<!-- =====Function y sus propiedades===== -->
<rdfs:Class rdf:ID="Function">
    <rdfs:label xml:lang="en">Function</rdfs:label>
    <rdfs:comment>Algorithm or formula performed to combine two or more metrics.</rdfs:comment>
    <rdfs:subClassOf rdf:resource="#RepositoryObject"/>
</rdfs:Class>
<rdf:Property rdf:ID="FunctionName">
    <rdfs:label xml:lang="en">Name</rdfs:label>

```

```

    <rdfs:domain rdf:resource="#Function"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="FunctionSpecification">
    <rdfs:label xml:lang="en">Specification</rdfs:label>
    <rdfs:domain rdf:resource="#Function"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<!-- =====Method y sus propiedades===== -->
<rdfs:Class rdf:ID="Method">
    <rdfs:label xml:lang="en">Method</rdfs:label>
    <rdfs:comment>Logical sequence of operations and possible heuristics, specified generically, for allowing
        the realisation of an activity description.</rdfs:comment>
    <rdfs:subClassOf rdf:resource="#RepositoryObject"/>
</rdfs:Class>
<rdf:Property rdf:ID="MethodName">
    <rdfs:label xml:lang="en">Name</rdfs:label>
    <rdfs:domain rdf:resource="#Method"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="MethodSpecification">
    <rdfs:label xml:lang="en">Specification</rdfs:label>
    <rdfs:domain rdf:resource="#Method"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="MethodReference">
    <rdfs:label xml:lang="en">Reference</rdfs:label>
    <rdfs:domain rdf:resource="#Method"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="AutomatedBy">
    <rdfs:label xml:lang="en">Automated by</rdfs:label>
    <rdfs:domain rdf:resource="#Method"/>
    <rdfs:range rdf:resource="#Instrument"/>
</rdf:Property>
<!-- =====Instrument y sus propiedades===== -->
<rdfs:Class rdf:ID="Instrument">
    <rdfs:label xml:lang="en">Method</rdfs:label>
    <rdfs:comment> ... </rdfs:comment>
    <rdfs:subClassOf rdf:resource="#RepositoryObject"/>
</rdfs:Class>
<rdf:Property rdf:ID="InstrumentName">
    <rdfs:label xml:lang="en">Name</rdfs:label>
    <rdfs:domain rdf:resource="#Instrument"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="InstrumentDescription">
    <rdfs:label xml:lang="en">Description</rdfs:label>
    <rdfs:domain rdf:resource="#Instrument"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<!-- =====SoftwareTool y sus propiedades===== -->
<rdfs:Class rdf:ID="SoftwareTool">
    <rdfs:label xml:lang="en">Method</rdfs:label>
    <rdfs:comment> ... </rdfs:comment>
    <rdfs:subClassOf rdf:resource="#Instrument"/>
</rdfs:Class>
<rdf:Property rdf:ID="SoftwareToolVersion">
    <rdfs:label xml:lang="en">Version</rdfs:label>

```

```

    <rdfs:domain rdf:resource="#SoftwareTool"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="SoftwareToolProvider">
    <rdfs:label xml:lang="en">Provider</rdfs:label>
    <rdfs:domain rdf:resource="#SoftwareTool"/>
    <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<!-- =====CalculationMethod y sus propiedades===== -->
<rdfs:Class rdf:ID="CalculationMethod">
    <rdfs:label xml:lang="en">CalculationMethod</rdfs:label>
    <rdfs:comment>The particular logical sequence of operations specified for allowing the realisation of
        a formula or indicator description by a calculation.</rdfs:comment>
    <rdfs:subClassOf rdf:resource="#Method"/>
</rdfs:Class>
<!-- =====MeasurementMethod y sus propiedades===== -->
<rdfs:Class rdf:ID="MeasurementMethod">
    <rdfs:label xml:lang="en">MeasurementMethod</rdfs:label>
    <rdfs:comment>The particular logical sequence of operations and possible heuristics specified for allowing
        the realisation of a metric description by a measurement.</rdfs:comment>
    <rdfs:subClassOf rdf:resource="#Method"/>
</rdfs:Class>
<rdf:Property rdf:ID="MeasurementMethodType">
    <rdfs:label xml:lang="en">Type</rdfs:label>
    <rdfs:comment>A measurement method can be objective or subjective.</rdfs:comment>
    <rdfs:domain rdf:resource="#MeasurementMethod"/>
    <rdfs:range rdf:resource="#TypeOfMeasurementMethod"/>
</rdf:Property>
<rdfs:Class rdf:ID="TypeOfMeasurementMethod">
    <rdfs:label xml:lang="en">Type of Measurement Method</rdfs:label>
    <rdfs:comment> ... </rdfs:comment>
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
</rdfs:Class>
<rdf:Description rdf:ID="OBJECTIVE">
    <rdf:type rdf:resource="#TypeOfMeasurementMethod"/>
    <rdfs:label xml:lang="en">Objective</rdfs:label>
</rdf:Description>
<rdf:Description rdf:ID="SUBJECTIVE">
    <rdf:type rdf:resource="#TypeOfMeasurementMethod"/>
    <rdfs:label xml:lang="en">Subjective</rdfs:label>
</rdf:Description>
<!-- =====ElementaryModel y sus propiedades===== -->
<rdfs:Class rdf:ID="ElementaryModel">
    <rdfs:label xml:lang="en">ElementaryModel</rdfs:label>
    <rdfs:comment>Algorithm or function with associated decision criteria that model
        an elementary indicator. </rdfs:comment>
    <rdfs:subClassOf rdf:resource="#Function"/>
    <rdfs:subClassOf rdf:resource="#ContextualEntity"/>
</rdfs:Class>
<rdf:Property rdf:ID="ElementaryModelInterpretedUsing">
    <rdfs:label xml:lang="en">Interpreted Using</rdfs:label>
    <rdfs:domain rdf:resource="#ElementaryModel"/>
    <rdfs:range rdf:resource="#DecisionCriteria"/>
</rdf:Property>
<rdf:Property rdf:ID="ElementaryModelInterprets">
    <rdfs:label xml:lang="en">Interprets</rdfs:label>
    <rdfs:domain rdf:resource="#ElementaryModel"/>
    <rdfs:range rdf:resource="#Metric"/>
</rdf:Property>

```

```

<!-- =====GlobalModel y sus propiedades===== -->
<rdfs:Class rdf:ID="GlobalModel">
  <rdfs:label xml:lang="en">Global Model</rdfs:label>
  <rdfs:comment>Algorithm or function with associated decision criteria that models
    a global indicator.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#RepositoryObject"/>
  <rdfs:subClassOf rdf:resource="#ContextualEntity"/>
</rdfs:Class>
<rdf:Property rdf:ID="GlobalModelInterpretedUsing">
  <rdfs:label xml:lang="en">Interpreted Using</rdfs:label>
  <rdfs:domain rdf:resource="#GlobalModel"/>
  <rdfs:range rdf:resource="#DecisionCriteria"/>
</rdf:Property>
<rdf:Property rdf:ID="GlobalModelOperator">
  <rdfs:label xml:lang="en">Operator</rdfs:label>
  <rdfs:domain rdf:resource="#GlobalModel"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<!-- =====DecisionCriteria y sus propiedades===== -->
<rdfs:Class rdf:ID="DecisionCriteria">
  <rdfs:label xml:lang="en">DecisionCriteria</rdfs:label>
  <rdfs:comment>Patterns used to determine the need for action or further investigation, or to describe the level
    of confidence in a given result.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#RepositoryObject"/>
  <rdfs:subClassOf rdf:resource="#ContextualEntity"/>
</rdfs:Class>
<rdf:Property rdf:ID="DecisionCriteriaRange">
  <rdfs:label xml:lang="en">Range</rdfs:label>
  <rdfs:domain rdf:resource="#DecisionCriteria"/>
  <rdfs:range rdf:resource="#DecisionCriterion"/>
</rdf:Property>
<!-- =====DecisionCriterion y sus propiedades===== -->
<rdfs:Class rdf:ID="DecisionCriterion">
  <rdfs:label xml:lang="en">Decision Criterion</rdfs:label>
  <rdfs:comment>Threshold values defining a level of acceptability.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#RepositoryObject"/>
</rdfs:Class>
<rdf:Property rdf:ID="DecisionCriterionName">
  <rdfs:label xml:lang="en">Name</rdfs:label>
  <rdfs:domain rdf:resource="#DecisionCriterion"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="DecisionCriterionDescription">
  <rdfs:label xml:lang="en">Description</rdfs:label>
  <rdfs:domain rdf:resource="#DecisionCriterion"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="DecisionCriterionLowerThreshold">
  <rdfs:label xml:lang="en">Lower Threshold</rdfs:label>
  <rdfs:domain rdf:resource="#DecisionCriterion"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
<rdf:Property rdf:ID="DecisionCriterionUpperThreshold">
  <rdfs:label xml:lang="en">Upper Threshold</rdfs:label>
  <rdfs:domain rdf:resource="#DecisionCriterion"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>
</rdf:RDF>

```


Bibliografía

- Abran, A., Bourque, P., Dupuis, R., Moore, J. W. & Tripp, L. L.** (2004), *Guide to the Software Engineering Body of Knowledge - SWEBOK*. IEEE Computer Society, URL http://www.swebok.org/ironman/pdf/SWEBOK_Guide_2004.pdf.
- Antunes, B., Correia, F. & Gomes, P.** (2011), *Context Capture in Software Development*. En *3rd Artificial Intelligence Techniques in Software Engineering Workshop, 7 October*, (arXiv.org - Cornell University Library), URL <http://arxiv.org/abs/1101.4101>.
- Araujo, R. M. D., Santoro, F. M., Brézillon, P., Borges, R. D. S., Gonçalves, M. & da Rosa, P.** (2003), *Context Models for Managing Collaborative Software Development Knowledge*. En *Proceedings of the KI-04 Workshop on Modeling and Retrieval of Context, Vol-114*, (CEUR-WS.org), URL <http://ceur-ws.org/Vol-114/>.
- Bailey, E., Card, D., Dean, J., Hall, F., Jones, C. & McGarry, B. L. J.** (2003), *Practical Software and Systems Measurement - A Foundation for Objective Project Management*. Department of Defense and US Army, version 4.0c edición.
- Bailey, J. W. & Basili, V. R.** (1981), *A Meta-model for software Development Resource Expenditures*. En *Proceedings of the Fifth Int'l Conference on Software Engineering*, (IEEE Press), ISBN 0-89791-146-6, págs. 107–116.
- Baldauf, M., Dustdar, S. & Rosenberg, F.** (2007), *A Survey on Context-Aware Systems*. En *International Journal of Ad Hoc and Ubiquitous Computing*, (Inderscience Enterprises Ltd.), tomo 2(4); págs. 263–277.
- Barcellos, M. P., Falbo, R. d. A. & Dal Moro, R.** (2010), *A Well-Founded Software Measurement Ontology*. En *Proceeding of the 2010 conference on Formal Ontology in Information Systems*, (IOS Press, Amsterdam, The Netherlands), ISBN 978-1-60750-534-1, págs. 213–226.
- Basili, V., Heidrich, J., Lindvall, M., Münch, J., Regardie, M., Rombach, D., Seaman, C. & Trendowicz, A.** (2007), *Bridging the Gap between Business Strategy and Software Development*. En *ICIS 2007 Proceedings*, (AIS Electronic Library), págs. 1–16.
- Basili, V. R., Caldiera, G. & Rombach, H. D.** (1994), *The Goal Question Metric Approach*. En *Encyclopedia of Software Engineering*, (Wiley), págs. 528–532.

- Basili, V. R. & Rombach, H. D.** (1988), *The TAME Project: Towards Improvement-Oriented Software Environments*. Informe Técnico CS-TR-1983, Institute for Advanced Computer Studies - Department of Computer Science - University of Maryland.
- Becker, P., Lew, P. & Olsina, L.** (2012), *Specifying process views for a measurement, evaluation, and improvement strategy*. En *Advances in Software Engineering*, (Hindawi Publishing Corp., New York, NY, United States), tomo 2012; pág. 28, ISSN 1687-8655, doi:10.1155/2012/949746, URL <http://dx.doi.org/10.1155/2012/949746>.
- Becker, P., Molina, H. & Olsina, L.** (2010), *Measurement and Evaluation as quality driver*. En *Journal Ingénierie des Systèmes d'Information, Special Issue "Quality of Information Systems"*, (Hermès-Science & Lavoisier), tomo 15(6); págs. 33–62.
- Belotti, R., Decurtins, C., Grossniklaus, M., Norrie, M. C. & Palinginis, A.** (2004), *Modelling Context for Information Environments*. En *Workshop on Ubiquitous Mobile Information and Collaboration Systems (UMICS)*, (16th International Conference on Advanced Information Systems Engineering - CAiSE 2004, Riga, Latvia), págs. 43–56.
- Berander, P., Damm, L., Eriksson, J., Gorschek, T., Henningsson, K., Jonsson, P., Kagstrom, S., Milicic, D., Martensson, F., Ronkko, K., Tomaszewski, P., Lundberg, L., Mattsson, M. & Wohlin, C.** (2005), *Software quality attributes and trade-offs*, (Blekinge Institute of Technology).
- Berners-Lee, T.** (1998), *Relational Databases on the Semantic Web*, World Wide Web Consortium (W3C), URL <http://www.w3.org/DesignIssues/RDB-RDF.html>.
- Berners-Lee, T., Fielding, R. & Masinter, L.** (2005), *RFC 3986: Uniform Resource Identifier (URI): Generic Syntax*, Network Working Group, URL <http://www.ietf.org/rfc/rfc3986.txt>.
- Bevan, N.** (1999), *Quality in Use: Meeting User Needs for Quality*. En *Journal of System and Software*, (Elsevier), tomo 49(1); págs. 89–96.
- Bizer, C., Cyganiak, R., Garbers, J., Maresch, O. & Becker, C.** (2009), *The D2RQ Platform v0.7 - Treating Non-RDF Relational Databases as Virtual RDF Graphs*, URL <http://www4.wiwiss.fu-berlin.de/bizer/d2rq/spec/>.
- Braungarten, R.** (2007), *The SMPI model: A stepwise process model to facilitate software measurement process improvement along the measurement paradigms*. Tesis Doctoral, Fakultät für Informatik der Otto-von-Guericke-Universität Magdeburg.
- Brézillon, P.** (2002), *Modeling and Using Context: Past, Present and Future*. Informe técnico, Laboratoire d'Informatique de Paris 6, URL citeseer.ist.psu.edu/549676.html.
- Briand, L. C., Morasca, S. & Basili, V. R.** (2002), *An Operational Process for Goal-Driven Definition of Measures*. En *IEEE Transactions on Software Engineering*, (IEEE Computer Society), tomo 28(12); págs. 1106–1125.
- Calero, C., Ruiz, J. & Piattini, M.** (2004), *A Web Metrics Survey Using WQM*. En *Lecture Notes in Computer Science*, (Springer), tomo 3140; págs. 147–160.
- Card, D. & Maclver, R.** (2003), *Applying PSM to Enterprise Measurement*. Informe técnico, Software Productivity Consortium.
- Card, D. N.** (2000), *A Practical Framework For Software Measurement And Analysis*. Informe técnico, Auerbach Software Management Strategies.

-
- Carleton, A. D., Park, R. E., Goethert, W. B., Florac, W. A., Bailey, E. K. & Pfleeger, S. L.** (1992), *Software Measurement for DoD Systems: Recommendations for Initial Core Measures*. Informe Técnico CMU/SEI-92-TR-19, ESC-TR-92-019, Software Engineering Institute.
- Chaves, A. P., Huzita, E. H. M., Vieira, V. & Steinmacher, I.** (2010), *A Context Conceptual Model for a Distributed Software Development Environment*. En *Proceedings of the 22nd International Conference on Software Engineering & Knowledge Engineering (SEKE'2010), Redwood City, San Francisco Bay, CA, USA, July 1 - July 3, 2010*, (Knowledge Systems Institute Graduate School), págs. 437–442.
- Chen, G. & Kotz, D.** (2000), *A Survey of Context-Aware Mobile Computing Research*. Informe técnico, Department of Computer Science, Dartmouth College.
- CMMI Product Team** (2006), *CMMI for Development, Version 1.2*. Software Engineering Institute.
- Covella, G. J. & Olsina, L. A.** (2006), *Assessing quality in use in a consistent way*. En *ICWE '06: Proceedings of the 6th international conference on Web engineering*, (ACM, New York, NY, USA), ISBN 1-59593-352-2, págs. 1–8.
- Cruzes, D., Mendonça, M. G., Basili, V., Shull, F. & Jino, M.** (2007), *Using Context Distance Measurement to Analyze Results across Studies*. En *ESEM'07: Proceedings of the First International Symposium on Empirical Software Engineering and Measurement*, (IEEE Computer Society, Washington, DC, USA), ISBN 0-7695-2886-4, págs. 235–244.
- David, J., Kögel, M. & Maalej, W.** (2007), *The conceptual model and specification of the context observer and history analyzer*. Informe Técnico 35111, TEAM Project.
- de Freitas Bulcão Neto, R. & da Graça Campos Pimentel, M.** (2005), *Toward a Domain-Independent Semantic Model for Context-Aware Computing*. En *Third Latin American Web Congress (LA-WEB) 2005.*, (IEEE CS Press), págs. 10–18.
- Debou, C., Kuntzmann-Combelles, A. & Rowe, A.** (1994), *A Quantitative Approach to Software Process Management*. En *Proceedings of the Second International Software Metrics Symposium*, (IEEE Computer Society), págs. 26–34.
- Dey, A. K.** (2000), *Providing Architectural Support for Building Context-Aware Applications*. Tesis Doctoral, Georgia Institute of Technology, Doctor of Philosophy in Computer Science.
- Dey, A. K.** (2001), *Understanding and Using Context*. En *Personal Ubiquitous Computing*, (Springer-Verlag, London, UK), tomo 5(1); págs. 4–7, ISSN 1617-4909.
- Dey, A. K. & Abowd, G. D.** (1999), *Towards a Better Understanding of Context and Context-Awareness*. En *HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, (Springer-Verlag, London, UK), ISBN 3-540-66550-1, págs. 304–307.
- Dos Santos, V. V.** (2008), *CEManTIKA: A Domain-Independent Framework for Designing Context-Sensitive Systems*. Tesis Doctoral, Universidade Federal de Pernambuco.
- Dourish, P.** (2004), *What we talk about when we talk about context*. En *Personal Ubiquitous Computing*, (Springer-Verlag, London, UK), tomo 8(1); págs. 19–30, ISSN 1617-4909.
- Dujmovic, J.** (1996), *A Method For Evaluation And Selection Of Complex Hardware And Software Systems*. En *CMG 96 Proceedings*, (The Computer Measurement Group, Inc.), págs. 368–378.
-

- Dujmovic, J.** (2007), *Continuous Preference Logic for System Evaluation*. En IEEE Transactions on Fuzzy Systems, (IEEE Computational Intelligence Society), tomo 15(6); págs. 1082–1099, ISSN 1063-6706.
- Ebert, C. & Dumke, R.** (2007), *Software Measurement: Establish - Extract - Evaluate - Execute*, (Springer-Verlag Berlin Heidelberg), ISBN 978-3-540-71648-8.
- Fenton, N. E. & Pfleger, S. L.** (1997), *Software metrics: a rigorous and practical approach*, (Boston, EUA : PWS), ISBN 0-534-95425-1.
- Florac, W. A.** (1992), *Software Quality Measurement: A Framework for Counting Problems and Defects*. Informe Técnico CMU/SEI-92-TR-022, Software Engineering Institute.
- Florac, W. A., Park, R. E. & Carleton, A. D.** (1997), *Practical Software Measurement: Measuring for Process Management and Improvement*. Guidebook CMU/SEI-97-HB-003, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA 15213, URL <http://www.sei.cmu.edu/library/abstracts/reports/97hb003.cfm>.
- Fuchs, F., Hochstatter, I., Krause, M. & Berger, M.** (2005), *A Metamodel Approach to Context Information*. En *Proceedings of the 3rd Int'l Conf. on Pervasive Computing and Communications Workshops*, PerCom 2005, (IEEE Computer Society), ISBN 0-7695-2300-5, págs. 8–14.
- García, F., Bertoa, M. F., Calero, C., Vallecillo, A., Ruíz, F., Piattini, M. & Genero, M.** (2006), *Towards a consistent terminology for software measurement*. En *Information and Software Technology* (), tomo 48(8); págs. 631–644.
- García, F., Ruíz, F., Bertoa, M. F., Calero, C., Genero, M., Olsina, L., Martín, M., Quer, C., Tondori, N., Abrahao, S., Vallecillo, A. & Piattini, M.** (2004), *Una Ontología de la Medición del Software*. Technical Report UCLM DIAB-04-02-2, Computer Science Department - University of Castilla-La Mancha, Spain.
- García, F., Ruiz, F., Calero, C., Bertoa, M. f., Vallecillo, A., Mora, B. & Piattini, M.** (2009), *Effective use of ontologies in software measurement*. En *The Knowledge Engineering Review*, (Cambridge University Press, New York, NY, USA), tomo 24(1); págs. 23–40, ISSN 0269-8889.
- García, F., Serrano, M., Cruz-Lemus, J., Ruíz, F. & Piattini, M.** (2007), *Managing software process measurement: A metamodel-based approach*. En *Information Sciences*, (Elsevier Science Inc., New York, NY, USA), tomo 177(12); págs. 2570–2586, ISSN 0020-0255.
- Goethert, W. & Fisher, M.** (2003), *Deriving Enterprise-Based Measures Using the Balanced Scorecard and Goal-Driven Measurement Techniques*. Technical Note CMU/SEI-2003-TN-024, Software Engineering Institute.
- Gong, L.** (2005), *Contextual Modeling and Applications*. En *IEEE International Conference on Systems, Man and Cybernetics*, (IEEE Systems, Man, and Cybernetics Society), ISBN 0780392981, págs. 381–386.
- Gresse, C., Hoisl, B. & Wüst, J.** (1995), *A process model for QGM-based measurement*. Informe Técnico STTI-95-04-E, The Software Technology Transfer Initiative Kaiserslautern (STTI-KL).
- Grossniklaus, M. & Norrie, M. C.** (2007), *An object-oriented version model for context-aware data management*. En *WISE'07: Proceedings of the 8th international conference on Web information systems engineering*, (Springer-Verlag, Berlin, Heidelberg), ISBN 3-540-76992-7, págs. 398–409.

-
- Gu, T., Wang, X. H., Pung, H. K. & Zhang, D. Q.** (2004), *An Ontology-based Context Model in Intelligent Environments*. En *Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference*, (Society for Modeling & Simulation International), págs. 270–275.
- Henrich, A. & Morgenroth, K.** (2003), *Supporting Collaborative Software Development by Context-Aware Information Retrieval Facilities*. En *Proceedings of the 14th International Workshop on Database and Expert Systems Applications*, (IEEE Computer Society, Washington, DC, USA), ISBN 0-7695-1993-8, págs. 249–254.
- Huang, W. & Tao, T.** (2004), *Adding Context-awareness to Knowledge Management in Modern Enterprises*. En *Second IEEE International Conference On Intelligent Systems*, (IEEE Computer Society), págs. 393–398.
- IEEE Std. 1061** (2004), *IEEE Standard 1061 for a Software Quality Metrics Methodology*. IEEE Standard.
- ISO/IEC 14598-1** (1999), *ISO/IEC 14598-1:1999 – Information technology – Software product evaluation – Part 1: General Overview*. Geneva, Switzerland.
- ISO/IEC 15504** (2004), *ISO/IEC 15504:2004 – Information technology – Process assessment*. Geneva, Switzerland.
- ISO/IEC 15939** (2001), *ISO/IEC 15939:2001 - Software Engineering - Software Measurement Process*. Geneva, Switzerland.
- ISO/IEC 25000** (2005), *ISO/IEC 25000 Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Guide to SQuaRE*.
- ISO/IEC 25010** (2011), *ISO/IEC 25010 Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Software and quality in use models*.
- ISO/IEC 25012** (2008), *ISO/IEC 25012 Software engineering – Software Quality Requirements and Evaluation (SQuaRE) – Data Quality Models*.
- ISO/IEC 25020** (2007), *ISO/IEC 25020 Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Measurement reference model and guide*.
- ISO/IEC 9126-1** (2001), *ISO/IEC 9126-1 Software Engineering- Product Quality. Part 1: Quality Model*.
- ISO/IEC 99:1993** (1993), *International vocabulary of basic and general terms in metrology (VIM)*.
- Java API DOM-HTML** (2003), *Java API for Document Object Model (DOM) - HTML*, Oracle, URL <http://docs.oracle.com/javase/1.4.2/docs/guide/plugin/dom/org/w3c/dom/html/package-summary.html>.
- Java API DOM-XML** (2003), *Java API for Document Object Model (DOM) - XML*, Oracle, URL <http://docs.oracle.com/javase/1.4.2/docs/api/org/w3c/dom/package-summary.html>.
- Jendrock, E., Evans, I., Gollapudi, D., Haase, K., Oliveira, W. M. & Srivathsa, C.** (2012), *The Java EE 6 Tutorial*, (Oracle), capítulo 15. Java Servlet Technology. URL <http://docs.oracle.com/javaee/6/tutorial/doc/>.
- Juran, J. & Godfrey, A. B.** (1999), *Juran's Quality Handbook*, (McGraw-Hill), ISBN 0-07-034003-X.
-

- Kaltz, J. W., Ziegler, J. & Lohmann, S.** (2005), *Context-aware Web Engineering: Modeling and Applications*. En RIA Revue d'Intelligence Artificielle, Special Issue on Applying ContextManagement, (Lavoisier), tomo 19(3); págs. 439–458, ISSN 0992499X.
- Kan, S. H.** (2002), *Metrics and Models in Software Quality Engineering*, (Addison Wesley), 2 edición, ISBN 0-201-72915-6.
- Kashyap, V. & Sheth, A.** (1997), *Cooperative Information Systems: Trends and Directions*, (Academic Press), capítulo Semantic Heterogeneity in Global Information Systems: The Role of Metadata, Context and Ontologies. 1 edición, ISBN 0-125-44910-0, págs. 139–178.
- Kessler, C.** (2007), *Similarity Measurement in Context*. En *6th International and Interdisciplinary Conference, CONTEXT 2007, Roskilde, Denmark, August 20-24*, tomo 4635 de *Lecture Notes in Artificial Intelligence* (**Kokinov, B., Richardson, D. & Roth-Berghofer, L., Th.R. and Vieu, eds.**), (Springer-Verlag Berlin Heidelberg), ISBN 978-3-540-74254-8, págs. 277–290.
- Kitchenham, B.** (1987), *Towards a constructive quality model: part 1: software quality modelling, measurement and prediction*. En *Software Engineering Journal*, (Michael Faraday House, Herts, UK, UK), tomo 2(4); págs. 105–113, ISSN 0268-6961.
- Kitchenham, B., Hughes, R. T. & Linkman., S. G.** (2001), *Modeling Software Measurement Data*. En *IEEE Transactions On Software Engineering*, (IEEE Computer Society), tomo 27(9); págs. 788–804.
- Kitchenham, B., Jeffery, D. R. & Connaughton, C.** (2007), *Misleading Metrics and Unsound Analyses*. En *IEEE Software*, (IEEE Computer Society, Los Alamitos, CA, USA), tomo 24(2); págs. 73–78, ISSN 0740-7459.
- Kitchenham, B., Pfleeger, S. L. & Fenton, N.** (1995), *Towards a Framework for Software Measurement Validation*. En *IEEE Transactions On Software Engineering*, (IEEE Computer Society), tomo 21(12); págs. 929–944.
- Klemke, R.** (2002), *Modelling Context in Information Brokering Processes*. Tesis Doctoral, Faculty of Mathematics, Computer Science and Natural Sciences at the RWTH Aachen.
- Laird, L. M. & Brennan, M. C.** (2006), *Software Measurement and Estimation: A Practical Approach*, (IEEE Computer Society - Wiley Interscience), ISBN 978-0471676225.
- Lawler, J. & Kitchenham, B.** (2003), *Measurement Modeling Technology*. En *IEEE Software*, (IEEE Computer Society), tomo 20(3); págs. 68–75, ISSN 0740-7459.
- Lew, P., Olsina, L., Becker, P. & Zhang, L.** (2011), *An integrated strategy to systematically understand and manage quality in use for web applications*. En *Requirements Engineering*, (Springer London), tomo 16; págs. 1–32, ISSN 0947-3602.
- Lew, P., Olsina, L. & Zhang, L.** (2010), *Quality, Quality in Use, Actual Usability and User Experience as Key Drivers for Web Application Evaluation*. En *Proceedings of 10th International Conference on Web Engineering, ICWE 2010, Vienna, Austria, July 5–9*, tomo 6189 de *Lecture Notes in Computer Science* (**Benatallah, B., Casati, F., Kappel, G. & Rossi, G., eds.**), (Springer-Verlag Berlin, Heidelberg), ISBN 978-3-642-13910-9, págs. 218–232.
- Lonsdale, P. & Beale, R.** (2004), *Towards a dynamic process model of context*. En *Proceedings of Ubicomp Workshop on Advanced Context Modeling, Reasoning and Management*, (UbiComp).

-
- Maalej, W., Happel, H.-J., Ntioudis, S. & David, J.** (2006), *State-of-the-art regarding personalisation and contextualisation in software engineering*. Informe técnico, TEAM Project - Information Society Technology Programme, Framework Programme 6.
- Martín, M. & Olsina, L.** (2003), *Towards an Ontology for Software Metrics and Indicators as the Foundation for a Cataloging Web System*. En *Proceedings of the First Conference on Latin American Web Congress*, (IEEE Computer Society, Washington, DC, USA), ISBN 0-7695-2058-8, págs. 103–117.
- McAndrews, D. R.** (1993), *Establishing a Software Measurement Process*. Informe Técnico CMU/SEI-93-TR-16, Software Engineering Institute, URL <http://www.sei.cmu.edu/library/abstracts/reports/93tr016.cfm>.
- Molina, H. & Olsina, L.** (2007a), *Soporte de Información Contextual en un Marco de Medición y Evaluación*. En *Conferencia Iberoamericana de Software Engineering (CIBSE)* (), págs. 77–90.
- Molina, H. & Olsina, L.** (2007b), *Towards the Support of Contextual Information to a Measurement and Evaluation Framework*. En *6th International Conference on Quality of Information and Communications Technology, QUATIC 2007*, (IEEE Computer Society), ISBN 0-7695-2948-8, págs. 154–166.
- Molina, H. & Olsina, L.** (2008), *Assessing Web Applications Consistently: A Context Information Approach*. En *Eighth International Conference on Web Engineering, ICWE '08.*, (IEEE Computer Society), págs. 224–230.
- Molina, H., Olsina, L. & Rossi, G.** (2010), *Context-Based Recommendation Approach for Measurement and Evaluation Projects*. En *Journal of Software Engineering and Applications*, (Scientific Research Publishing), tomo 3(12); págs. 1089–1106, ISSN 1945-3116.
- Molina, H., Papa, F., de los Angeles Martín, M. & Olsina, L.** (2004), *Semantic Capabilities for the Metrics and Indicators Cataloging Web System*. En *Engineering Advanced Web Applications* (Matera, M. & Comai, S., eds.), (Rinton Press Inc.), ISBN 1-58949-046-0, págs. 97–109.
- Munson, J. C.** (2003), *Software Engineering Measurement*, (Auerbach Publications), ISBN 0849315034.
- Olsina, L.** (1999), *Metodología Cuantitativa para la Evaluación y Comparación de la Calidad de Sitios Web*. Tesis Doctoral, Facultad de Ciencias Exactas, Universidad Nacional de La Plata.
- Olsina, L., Covella, G. & Rossi, G.** (2006), *Web Quality*. En *Web Engineering* (Mendes, E. & Mosley, N., eds.), (Springer Berlin Heidelberg), ISBN 978-3-540-28218-1, págs. 109–142.
- Olsina, L., Garrido, A., Rossi, G., Distante, D. & Canfora, G.** (2008a), *Web application evaluation and refactoring: a quality oriented improvement approach*. En *Journal of Web Engineering*, (Rinton Press, Paramus, NJ), tomo 7(4); págs. 258–280, ISSN 1540–9589.
- Olsina, L., Lew, P., Dieser, A. & Rivera, B.** (2011), *Using Web Quality Models and a Strategy for Purpose-Oriented Evaluations*. En *Journal of Web Engineering*, (Rinton Press), tomo 10(4); págs. 316–352.
- Olsina, L. & Martín, M.** (2004), *Ontology for Software Metrics and Indicators*. En *Journal of Web Engineering*, (Rinton Press), tomo 2(4); págs. 262–281, URL <http://www.rintonpress.com/journals/jweonline.html#v2n4>.
-

- Olsina, L., Molina, H. & Papa, F.** (2008b), *Web Engineering: Modelling and Implementing Web Applications*, (Springer London), capítulo 13. How to Measure and Evaluate Web Applications in a Consistent Way. ISBN 978-1-84628-922-4, págs. 385–420.
- Olsina, L., Papa, F. & Molina, H.** (2005), *Organization-Oriented Measurement and Evaluation Framework for Software and Web Engineering Projects*. En *Proceedings of the International Conference on Web Engineering (ICWE) 2005*, tomo 3579 de *Lecture Notes in Computer Science* (**Lowe, D. & Gaedke, M.**, eds.), (Springer-Verlag, Berlin Heidelberg), págs. 42–52.
- Olsina, L., Papa, F. & Molina, H.** (2008c), *Ontological support for a measurement and evaluation framework*. En *International Journal of Intelligent Systems*, (Wiley, New York, NY, USA), tomo 23(12); págs. 1282–1300, ISSN 0884-8173.
- Olsina, L., Papa, M., Souto, M. & Rossi, G.** (2001), *Providing Automated Support for the Web Quality Evaluation Methodology*. En *Fourth Workshop on Web Engineering at the 10th International WWW Conference*, (International World Wide Web Conference Committee), ISBN 1-58113-348-0, págs. 1–11. Hong Kong.
- Olsina, L. & Rossi, G.** (2002), *Measuring Web Application Quality with WebQEM*. En *IEEE Multimedia*, (IEEE Computer Society Press, Los Alamitos, CA, USA), tomo 9(4); págs. 20–29, ISSN 1070-986X.
- Olsina, L., Sassano, R. & Mich, L.** (2009), *Towards the Quality for Web 2.0 Applications*. En *Proceedings of the 8th International Workshop on Web-Oriented Software Technologies (IWWOST'2009)*, tomo 493, (CEUR-WS.org), págs. 3–15.
- OMG-MOF** (2012), *MetaObject Facility (MOF)*, Object Management Group, URL <http://www.omg.org/mof/>.
- OMG-UML** (2012), *The Unified Modeling Language (UML)*, Object Management Group, URL <http://uml.org/>.
- Park, R. E., Goethert, W. B. & Florac, W. A.** (1996), *Goal-Driven Software Measurement –A Guidebook*. Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA 15213, URL <http://www.sei.cmu.edu/library/abstracts/reports/96hb002.cfm>.
- PMI, P. M. I.** (2000), *A Guide to the Project Management Body of Knowledge (PMBOK® guide)*, (Project Management Institute, Newton Square, Pennsylvania USA), 2000 edición, ISBN 1-880410-25-7.
- Pressman, R. S.** (2001), *Software Engineering: A Practitioner's Approach*, (McGraw-Hill Higher Education), 5th edición, ISBN 0072496681.
- Rodríguez, M., Genero, M., Torre, D., Blasco, B. & Piattini, M.** (2010), *A Methodology for Continuous Quality Assessment of Software Artefacts*. En *QSIC '10: Proceedings of the 2010 10th International Conference on Quality Software*, (IEEE Computer Society, Washington, DC, USA), ISBN 978-0-7695-4131-0, págs. 254–261.
- RQL-ICS-FORTH** (2008), *The RDF Query Language (RQL)*, FORTH Institute of Computer Science, URL <http://139.91.183.30:9090/RDF/RQL/>.
- Sahoo, S. S., Halb, W., Hellmann, S., Idehen, K., Jr, T. T., Auer, S., Sequeda, J. & Ezzat, A.** (2009), *A Survey of Current Approaches for Mapping of Relational Databases to RDF*, W3C, URL http://www.w3.org/2005/Incubator/rdb2rdf/RDB2RDF_SurveyReport.pdf.

-
- Schubert, P. & Dettling, W.** (2002), *Extended Web Assessment Method (EWAM) - Evaluation of e-commerce applications from the customer's viewpoint*. En *Proceedings of the 35th Annual Hawaii International Conference on System Sciences, HICSS 2002*, (IEEE Computer Society), pág. 175b.
- Schwarz, S.** (2005), *A context model for personal knowledge management*. En *Modeling and Retrieval of Context, Second International Workshop, MRC 2005*, (Springer), págs. 18–33.
- Seffah, A., Donyaee, M., Kline, R. & Padda, H.** (2006), *Usability measurement and metrics: A consolidated model*. En *Software Quality Journal*, (Springer Netherlands), tomo 14; págs. 159–178, ISSN 0963-9314.
- Sesame-API** (2006), *Sesame API release 1.2.7*, openRDF, URL <http://www.openrdf.org/doc/api/index.html>.
- Seshadri, G.** (1999), *Understanding JavaServer Pages Model 2 architecture - Exploring the MVC design pattern*, JavaWorld.com, URL <http://www.javaworld.com/javaworld/jw-12-1999/jw-12-ssj-jspmvc.html>.
- Stefani, A., Xenos, M. & Stavrinoudis, D.** (2003), *Modelling e-commerce systems' quality with belief networks*. En *Virtual Environments, Human-Computer Interfaces and Measurement Systems, 2003. VECIMS '03. 2003 IEEE International Symposium on*, (IEEE Computer Society), págs. 13–18.
- Stöttinger, M.** (2004), *Context-Awareness in industrial environments*, Tesis de Grado (Hagenberg, Austria), URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.117.3027&rep=rep1&type=pdf>.
- Strang, T. & Linnhoff-Popien, C.** (2004), *A Context Modeling Survey*. En *Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004 - The Sixth International Conference on Ubiquitous Computing, Nottingham, UK*, (UbiComp.org), págs. 34–41.
- Strang, T., Linnhoff-Popien, C. & Frank, K.** (2003), *CoOL: A Context Ontology Language to enable Contextual Interoperability*. En *Proceedings of 4th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems (DAIS2003)*, (Springer, Berlin), ISBN 9783540205296, págs. 236–247.
- Theodorakis, M.** (2001), *Contextualization: An Abstraction Mechanism for Information Modeling*. Doctoral dissertation, University of Crete, Department of Computer Science.
- van Solingen, R. & Berghout, E.** (1997), *Improvement by goal-oriented measurement*. En *Proceedings of the European Software Engineering Process Group conference (EUROPEAN SEPG 97), June 16-20*, Software Engineering Institute, (Carnegie-Mellon University), URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.96.5030&rep=rep1&type=pdf>.
- van Solingen, R. & Berghout, E.** (1999), *The Goal/Question/Metric Method – A Practical Guide for Quality Improvement of Software Development*, (McGraw-Hill Publishing Company), ISBN 007-709553-7.
- W3C-DOM** (1998), *Document Object Model (DOM) Level 1 Specification Version 1.0*, W3C, URL <http://www.w3.org/TR/REC-DOM-Level-1>.
- W3C-OWL** (2004), *OWL Web Ontology Language Overview*, W3C, URL <http://www.w3.org/TR/owl-features/>.
-

- W3C-R2RML** (2011), *R2RML: RDB to RDF Mapping Language*, W3C, URL <http://www.w3.org/TR/r2rml/>.
- W3C-RDB2RDF-Group** (2011), *RDB2RDF Working Group*, W3C, URL <http://www.w3.org/2001/sw/rdb2rdf/>.
- W3C-RDB2RDF-Workshop** (2007), *W3C Workshop on RDF Access to Relational Databases*, W3C, Cambridge, MA, USA, URL <http://www.w3.org/2007/03/RdfRDB/>.
- W3C-RDBDirectMapping** (2011), *A Direct Mapping of Relational Data to RDF*, W3C, URL <http://www.w3.org/TR/rdb-direct-mapping/>.
- W3C-RDF** (2004), *Resource Description Framework (RDF)*, W3C, URL <http://www.w3.org/RDF/>.
- W3C-RDFS** (2004), *RDF Vocabulary Description Language 1.0: RDF Schema*, W3C, URL <http://www.w3.org/TR/rdf-schema/>.
- W3C-SPARQL** (2008), *SPARQL Query Language for RDF*, W3C, URL <http://www.w3.org/TR/rdf-sparql-query/>.
- W3C-XML** (2012), *Extensible Markup Language (XML)*, W3C, URL <http://www.w3.org/XML/>.
- Wang, X. H., Zhang, D. Q., Gu, T. & Pung, H. K.** (2004), *Ontology Based Context Modeling and Reasoning using OWL*. En *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004.*, (IEEE Computer Society), págs. 18–22.
- XMLC** (2004), *Enhydra XMLC: Open Source Java/XML Presentation Compiler*, Object Web, URL <http://xmlc.ow2.org/doc/>.
- Zimmermann, A., Lorenz, A. & Oppermann, R.** (2007), *An Operational Definition of Context*. En *Modeling and Using Context - 6th International and Interdisciplinary Conference, CONTEXT 2007*, tomo 4635 de *Lecture Notes in Computer Science* (**Kokinov, B., Richardson, D. C., Roth-Berghofer, T. R. & Vieu, L.**, eds.), (Springer-Verlag, Berlin, Heidelberg), págs. 558–571.
- Zuse, H.** (1998), *A Framework of Software Measurement*, (Walter de Gruyter & Co., Hawthorne, NJ, USA), ISBN 3110155877.