

MARCO CONCEPTUAL, METODOLÓGICO Y
TECNOLÓGICO PARA EL MODELADO Y EJECUCIÓN DE
LA ACTIVIDAD QUE SOPORTA EL DISEÑO DE LA
INTERFAZ DE USUARIO DE SISTEMAS
INTERACTIVOS, DENTRO DEL CONTEXTO DE
TD-MBUID Y A PARTIR DE LA INFRAESTRUCTURA DE
CIAF



MARIA LILI VILLEGAS RAMÍREZ

Tesis de Doctorado en Ciencias de la Electrónica

Director:

César Alberto Collazos Ordóñez
Doctor en Ciencias de la Computación

Co-Director:

William Joseph Giraldo Orozco
Doctor en Informática

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Sistemas
Línea de Investigación en Interacción Humano Computador
Popayán, abril 2017

MARIA LILI VILLEGAS RAMÍREZ

MARCO CONCEPTUAL, METODOLÓGICO Y
TECNOLÓGICO PARA EL MODELADO Y EJECUCIÓN DE
LA ACTIVIDAD QUE SOPORTA EL DISEÑO DE LA
INTERFAZ DE USUARIO DE SISTEMAS
INTERACTIVOS, DENTRO DEL CONTEXTO DE
TD-MBUID Y A PARTIR DE LA INFRAESTRUCTURA DE
CIAF

Tesis presentada a la Facultad de Ingeniería
Electrónica y Telecomunicaciones de la
Universidad del Cauca para la obtención del
Título de

Doctora en:
Ciencias de la Electrónica

Director:
Phd. César Alberto Collazos Ordóñez
Co-Director:
Phd. William Joseph Giraldo Orozco

Popayán
2017

A mi esposo Leandro y mi niño Alejandro.

A mis padres, Bernardo y Nancy.

A mi hermana, Natalia.

AGRADECIMIENTOS

“Demos gracias a los hombres
y a las mujeres que nos hacen felices,
ellos son los encantadores jardineros
que hacen florecer a nuestros espíritus”.

Will Rogers

Son muchas las personas que han aportado, directa o indirectamente, a la realización de esta tesis, a todas ellas les dedico este espacio.

En primer lugar, a mis directores, César y William, por su invaluable acompañamiento y profesionalismo durante todo el camino recorrido desde el inicio, hasta el final de esta investigación.

Este trabajo lo he dedicado a mi familia, pero igualmente, les agradezco toda su ayuda, toda su confianza y todo el soporte que me han brindado en todas las metas que he querido alcanzar.

Muy especialmente a Juan Carlos, por dedicar gran parte de su valioso tiempo al diseño de mis presentaciones en los congresos y al diseño de este libro.

A todos mis compañeros del grupo SINFOCI, Alexandra, Faber, Jaime, Jorge, Robinson, Oscar y Manuel, por compartir conmigo largas horas de trabajo, por su apoyo y amistad. Igualmente, a mis compañeros de estudio, Yenny, Andrés, y Wilson, del grupo IDIS, por tantas veces que compartimos, aún estando todos en la carrera de terminar nuestras tesis. Muy especialmente a Yenny, por su apoyo y amistad.

A Maritza y Leandro, por brindarme su amistad desde el primer día que empecé mis estudios en el doctorado.

A los doctores Juan y Josefina, que me acogieron en su grupo de investigación, durante mi estancia en la BUAP, México.

A Sandra, por toda su ayuda y su guía en Puebla, desde antes de viajar a México.

A los doctores Francisco y Patricia, por recibirme en su grupo de investigación durante mi estancia en la Universidad de Granada, España.

Finalmente, quiero agradecer a la Universidad del Quindío por brindarme la comisión de estudios, con todos sus beneficios. A los profesores del programa de Ing. de Sistemas y Computación por llevar a cabo mis labores mientras estuve estudiando. A Colciencias, por financiar mis estudios. A la Universidad del Cauca por aceptarme como uno de sus estudiantes y por financiar la asistencia a eventos y estancias. A la AUIP por brindarme la beca de movilidad para realizar la estancia en la Universidad de Granada.

RESUMEN

Los sistemas interactivos están diseñados para soportar interacciones complejas. El desarrollo de este tipo de sistemas involucra la captura de grandes volúmenes de información relacionada con la especificación de las interacciones entre usuarios y sistema y también entre usuarios, es decir, entre personas.

Esta tesis se ubica dentro del dominio de la Ingeniería de Software, el contexto es Interacción Humano Computador (HCI), específicamente el desarrollo de Sistemas Interactivos y más concretamente, el desarrollo de la Interfaz de Usuario, que se expresa mediante artefactos que son la integración de las contribuciones de múltiples expertos desde distintas perspectivas y puntos de vista. Donde finalmente, el producto es un sistema interactivo en el cual se ven reflejadas dichas contribuciones desde la Funcionalidad, la Colaboración, la Seguridad, el Marketing, la Accesibilidad, etc, en mayor o menor medida, según el propósito para el cual es implementado el sistema interactivo.

En este sentido, nos interesa comprender cómo esta información puede ser enlazada e integrada para que contribuya a la creación de artefactos que expresen la interfaz de usuario de una manera más formal y computacional. Para tal fin, se introduce la definición de un marco conceptual, metodológico y tecnológico que facilite a los desarrolladores la evaluación y definición de lenguajes y herramientas para el modelado y la ejecución de la actividad de sistemas interactivos. Este marco de desarrollo está soportado en la definición de una Taxonomía de la Actividad (TxA) que permite a los diseñadores, desagregar, descomponer o incluso adicionar información relacionada con la actividad desde las especificaciones de un sistema interactivo, para ser entendida de acuerdo al interés en una vista o punto de vista.

Se sustenta la utilidad de la TxA como marco de referencia en HCI, validando su aplicación como: 1. Instrumento de valoración para artefactos generados de la Ingeniería de Software. 2. Marco de evaluación de propuestas de notación para representar la actividad y 3. Framework para la definición de roadmaps para el desarrollo de sistemas interactivos.

ABSTRACT

Interactive systems are designed to support complex interactions. The development of this type of systems involves the capture of large volumes of information related to the specification of interactions between users and the system and also between users, that is, between people.

This thesis is located within the domain of Software Engineering, the context is Human Computer Interaction (HCI), specifically the development of Interactive Systems and more specifically, the development of the User Interface, which is expressed through artifacts that are integration of the contributions of multiple experts from different perspectives and points of view. Where, finally, the product is an interactive system in which these contributions are reflected from Functionality, Collaboration, Security, Marketing, Accessibility, etc., to a greater or lesser extent, according to the purpose for which the Interactive system.

In this sense, we are interested in understanding how this information can be linked and integrated to contribute to the creation of artifacts that express the user interface in a more formal and computational way. To this end, the definition of a conceptual, methodological and technological framework is introduced to provide developers with the assessment and definition of languages and tools for the modeling and execution of the interactive systems activity. This development framework is supported in the definition of an Activity Taxonomy (TxA) that allows designers to disaggregate, decompose or even add information related to the activity from the specifications of an interactive system, to be understood according to the interest in a view or point of view.

The usefulness of the TxA as a frame of reference in HCI is supported, validating its application as: 1. Valuation tool for artifacts generated from Software Engineering. 2. Framework of evaluation of proposals of notation to represent the activity and 3. Framework for the definition of roadmaps for the development of interactive systems.

ESTRUCTURA DEL DOCUMENTO

Esta memoria de investigación tiene la siguiente estructura:

Capítulo 1: Introducción

En este capítulo se presenta la conceptualización de la tesis, en él se aborda la problemática que originó el trabajo que aquí se presenta, así como el contexto en el que surge y se detallan los objetivos concretos que se plantean alcanzar como resultado del mismo.

Capítulo 2: Marco de referencia

En este capítulo se describe el marco de referencia en el que se fundamenta el desarrollo de esta investigación. El marco de referencia que se presenta en esta tesis consta de dos apartados principales: el marco teórico, y los antecedentes. El marco teórico es el resultado de una investigación preliminar sobre teorías, consolidadas en la literatura, que se están siguiendo como modelo de la realidad y, que se utiliza para tomar decisiones en el diseño y para ordenar y estructurar la investigación. El marco de antecedentes es un conjunto de antecedentes que sirven como referencia y que se encuentran en publicaciones o investigaciones recientes. Finalmente, se presentan las conclusiones en la que se analizan, por medio de clasificaciones, las propuestas estudiadas.

Capítulo 3: Taxonomía de la actividad para sistemas interactivos

En este capítulo se presenta la definición de la Taxonomía de la Actividad (TxA). Esta taxonomía se especifica en base al análisis de un conjunto de teorías, propuestas metodológicas, taxonomías y marcos de desarrollo, que se describen en el capítulo 2.

Capítulo 4: Usos de la Taxonomía de la actividad para sistemas interactivos

En este capítulo se presenta la aplicación de la Taxonomía de la Actividad propuesta, dentro del contexto de HCI. El objetivo de este capítulo es presentar la descripción de los usos de la TxA en los que se ha enfocado esta tesis, para luego describir la aplicación y valoración por cada uso mediante la definición de un caso específico.

Capítulo 5: Aplicación y valoración de los usos de la Taxonomía de la Actividad

El propósito de este capítulo es presentar, mediante casos de estudio, la validación de los usos que se describen para la Taxonomía de la Actividad en el capítulo 4.

Capítulo 6: Conclusiones y trabajo futuro

En este capítulo se exponen cuáles son las lecciones aprendidas como consecuencia del trabajo realizado y las principales contribuciones. Además se describen los trabajos futuros o líneas de investigación que quedan abiertas como consecuencia del mismo.

Anexo A: Evaluación de notaciones para el modelado de tareas - perfil de los encuestados.

Anexo B: Evaluación de modelado de tareas para sistemas interactivos - contexto

Anexo C: Selección de elementos de modelado de sistemas interactivos

Anexo D: Notación que soporta el modelado de la actividad para AMBUID

Anexo E: Metamodelo de la notación para AMBUID

Anexo F: Modelo de diseño de ejecución de máquinas de estado

TABLA DE CONTENIDO

RESUMEN

ABSTRACT

ESTRUCTURA DEL DOCUMENTO

CAPÍTULO 1: INTRODUCCIÓN.....	1
1.1 Motivación	4
1.2 Objetivos	7
1.2.1 <i>Objetivo general</i>	7
1.2.2 <i>Objetivo específico 1</i>	7
1.2.3 <i>Objetivo específico 2</i>	8
1.2.4 <i>Objetivo específico 3</i>	8
1.2.5 <i>Objetivo específico 4</i>	8
1.2.6 <i>Objetivo específico 5</i>	9
1.3 Metodología	9
1.3.1 <i>Definición del dominio y contexto</i>	10
1.3.2 <i>Construcción del marco de referencia</i>	11
1.3.4 <i>Formulación de la hipótesis</i>	11
1.3.3 <i>Determinación del problema</i>	12
1.3.5 <i>Propuesta de investigación</i>	12
1.3.6 <i>Validación de la propuesta</i>	12
1.3.7 <i>Análisis de resultados o conclusiones</i>	12
1.3.8 <i>Resumen</i>	12
CAPÍTULO 2: MARCO DE REFERENCIA.....	15
2.1 Fundamentos teóricos	17
2.1.1 <i>Ingeniería de software</i>	17
2.1.2 <i>Casos de uso</i>	27
2.1.3 <i>Workflow</i>	28
2.1.4 <i>Teoría de la actividad</i>	29
2.1.5 <i>Ingeniería semiótica</i>	31
2.1.6 <i>Análisis de la comunicación</i>	32
2.1.7 <i>Interacción humano computador (HCI)</i>	33
2.2 Antecedentes	35
2.2.1 <i>Modelado y ejecución de la actividad en sistemas interactivos, funcionalidad.....</i>	35

2.2.2 Modelado y ejecución de la actividad en sistemas interactivos, colaboración	41
2.2.3 Modelado y ejecución de la actividad en sistemas interactivos, IU	44
2.2.4 Taxonomías en ingeniería de software y HCI	54

CAPÍTULO 3: TAXONOMÍA DE LA ACTIVIDAD PARA SISTEMAS INTERACTIVOS . 59

3.1 Teoría de la actividad como base para la definición de la TxA	61
3.2 Clasificación de la actividad como base para la definición de la TxA	62
3.2.1 Aspecto o faceta	63
3.2.2 Nivel de abstracción	64
3.2.3 Nivel de granularidad	64
3.2.4 Tipo de actividad	65
3.2.5 Razones para elegir la clasificación del modelado de la actividad	66
3.3 Framework de Zachman como base para la definición de la TxA.....	66
3.4 Mapa de ruta para la definición de la TxA	67
3.4.1 Clasificación de propuestas mediante el uso de la clasificación de la actividad	67
3.4.2 Integración de propuestas para la definición del metamodelo base	72
3.4.3 Clasificación de propuestas según la taxonomía de Zachman	79
3.4.4 Clasificadores complementarios para cada celda de la TxA	83

CAPÍTULO 4: USOS DE LA TAXONOMÍA 95

4.1 Valoración de artefactos de la ingeniería de software	98
4.1.1 Validación de la usabilidad en casos de uso mediante la TxA	98
4.2 Evaluación de lenguajes de modelado	101
4.2.1 Evaluación de elementos de modelado en el desarrollo de sistemas interactivos .	101
4.2.2 Niveles de abstracción y modelado de la información	102
4.3 Definición de marcos de desarrollo de sistemas interactivos	104
4.3.1 Contexto desde los puntos de vista de la ingeniería de software y HCI	104
4.3.2 Componentes del roadmap	105

CAPÍTULO 5: APLICACIÓN Y VALORACIÓN DE LOS USOS 107

5.1 Valoración de artefactos de la ingeniería de software	109
5.1.1 Proceso de Incorporación	109
5.1.2 Equipo de expertos	110
5.1.3 Artefactos	110
5.1.4 Caso de estudio	111
5.2 Evaluación de lenguajes de modelado	119
5.2.1 Clasificación de elementos de modelado a partir de la TxA	119
5.2.2 Selección de elementos de modelado para la especificación de SI	124
5.3 Definición de marcos de desarrollo de sistemas interactivos	131
5.3.1 Adaptación del método de desarrollo de la interfaz de usuario	131

5.3.2 Contenido metodológico detallado de la propuesta.....	136
5.3.3 Modificaciones a la notación usada en TD-MBUID y definición de algunos elementos de modelado	143
5.3.4 Modificaciones a la herramienta de softwarea CIAT	147
5.3.5 Caso de estudio	159
CAPÍTULO 6: CONCLUSIONES Y TRABAJO FUTURO.....	171
6.1 Contexto	173
6.2 Resumen de las contribuciones	174
6.2.1 Contribuciones teóricas y conceptuales	174
6.2.2 Contribuciones a nivel metodológico	175
6.2.3 Contribuciones a nivel tecnológico.....	176
6.3 Trabajo futuro	176
6.4 Aportaciones principales	177
6.4.1 Publicaciones	178
6.4.2 Proyectos de investigación.	181
6.4.3 Pasantías y estancias de investigación	181
REFERENCIAS	183
ANEXO A: EVALUACIÓN DE NOTACIONES PARA EL MODELADO DE TAREAS - PERFIL DE LOS ENCUESTADOS	203
ANEXO B: EVALUACIÓN DE MODELADO DE TAREAS PARA SISTEMAS INTERAC- TIVOS - CONTEXTO	209
ANEXO C: SELECCIÓN DE ELEMENTOS DE MODELADO DE SISTEMAS INTERAC- TIVOS	225
ANEXO D: NOTACIÓN QUE SOPORTA EL MODELADO DE LA ACTIVIDAD PARA AMBUID	261
ANEXO E: METAMODELO DEFINIDO PARA AMBUID.....	269
ANEXO F: MODELO DE DISEÑO DE EJECUCIÓN DE MÁQUINAS DE ESTADO	281

LISTA DE FIGURAS

Figura 1.1: Concepción del trabajo de investigación (Giraldo, 2010)	10
Figura 2.1. Estructura de un Sistema de Actividad (Engeström, 1987)	29
Figura 2.2. Comparación Esquemática entre el Diseño Centrado en el Usuario y la Ingeniería Semiótica (Souza, 2005)	32
Figura 2.3. Workflow para la estructura de requisitos y Análisis de la Comunicación (González et al, 2011)	33
Figura 2.4. Base metodológica del análisis de la comunicación con un amplio método de desa- rrollo (España, 2011)	40
Figura 2.5. Etapas de la propuesta metodológica CIAM (Molina et al, 2006)	41
Figura 2.6. Arquitectura del Framework para la co-ejecución de los modelos de tareas y de sistema (Barboni, 2010)	46
Figura 2.7. Representación de una tarea en AMBOSS (Giese, 2008)	47
Figura 2.8. Notación de diseño centrado en el uso, extendida, para el modelado de la actividad (Constantine, 2006)	48
Figura 2.9. Vista lógica del diseño centrado en el uso con el modelado de la actividad (Cons- tantine, 2006)	49
Figura 2.10. Vista general de la propuesta metodológica para desarrollar UIs para sistemas de información basados en workflow (Garcia, 2010)	50
Figura 2.11. Mapa de ruta propuesto en MPiu+a (Granollers, 2004)	53
Figura 2.12. Mapa de ruta para el desarrollo de la interfaz de usuario en TD-MBUID (Giraldo, 2010)	54
Figura 3.1. Clasificación de la actividad según los aspectos o facetas del sistema (Giraldo, 2010)	63
Figura 3.2. Estructura de la Clasificación de la actividad en un aspecto. (Giraldo, 2010)	64
Figura 3.3. Mapa de ruta para la definición de laTxA	67
Figura 3.4. Metamodelo de la TA definido en (Georg, 2015)	74
Figura 3.5. Metamodelo de la Clasificación de la Actividad	74
Figura 3.6. Mappings entre la TA y la Clasificación de la Actividad	76
Figura 3.7. Capa de Integración para Análisis de Clasificadores	77
Figura 3.8. Relación de conceptos entre la TA, la CA y la capa de integración	78
Figura 3.9. Clasificadores para cada celda de la capa de integración	79
Figura 3.10. Metamodelo Base para definir la Estructura Taxonómica para la TxA	80
Figura 3.11. Metamodelo Detallado de la Taxonomía propuesta por Zachman, Nivel de Negocio	84

Figura 3.12. Metamodelo Detallado de la Taxonomía propuesta por Zachman, Nivel de Sistema	85
Figura 3.13. Fracción del Metamodelo de Zachman. Relaciones entre columnas a partir de la Función	86
Figura 3.14. Metamodelo de Zachman. Relaciones entre columnas a partir de la Función	87
Figura 3.15. Fracción del Metamodelo conceptual de la propuesta metodológica para desarrollo de IUs a partir de workflows (Guerrero, 2010)	88
Figura 3.16. Fracción del metamodelo de UML v.1.1. para Casos de Uso (OMG, 1999)	88
Figura 3.17. Representación de una tarea cooperativa en CIAF (Giraldo, 2010)	89
Figura 3.18. Estructura Taxonómica de la Actividad	91
Figura 4.1. Usos definidos para la taxonomía de la actividad propuesta	97
Figura 5.1: Esquema de incorporación de HCI en un proceso de desarrollo de software a través de la TxA y a partir de los Casos de Uso	110
Figura 5.2: Diagrama de CU "Recaudos a través de Cajero Electrónico Multifuncional"	111
Figura 5.3. Formulario resultante a partir de la captura del modelo mental de usuarios representativos	113
Figura 5.4: Clasificación de los elementos notacionales de UML y CIAN según la estructura de la Taxonomía de la Actividad, Nivel de abstracción Negocio	123
Figura 5.5: Clasificación de los elementos notacionales de UML y CIAN según la estructura de la Taxonomía de la Actividad, Nivel de Abstracción Sistema	123
Figura 5.6: Metodología de Trabajo que dirige el proceso de evaluación	124
Figura 5.7: Esquema de preparación de la evaluación	125
Figura 5.8: Nivel de titulación de las personas encuestadas	127
Figura 5.9. Nivel de formación en HCI de las personas encuestadas	127
Figura 5.10. Nivel de experiencia vs. nivel de aplicación de técnicas de HCI	128
Figura 5.11. Elementos seleccionados para los clasificadores en el Nivel de Negocio	128
Figura 5.12. Elementos seleccionados para los clasificadores en el Nivel de Sistema	129
Figura 5.13. Imagen que resume los elementos relacionados a la HCI (Granollers, 2004)	130
Figura 5.14: Flujo de desarrollo de la interfaz de usuario en TD-MBUID	131
Figura 5.15. Mapa de ruta propuesto, especificación en SPEM	132
Figura 5.16: Mapa de ruta propuesto	134
Figura 5.17. Abstracciones canónicas presentadas por Constantine	138
Figura 5.18. Estructura del EMTR (Giraldo, 2000)	140
Figura 5.19. Relación entre los componentes EMTR y UML Semantics (Villegas, 2009)	141
Figura 5.20. Aspecto de las tareas de Inter-Acción de TD-MBUID. Propuesta de modificación, Tarea interactiva de negocio	144
Figura 5.21. Detalle de presentación actual y presentación nueva, Inter-acción de negocio	145
Figura 5.22. Detalle de presentación actual y presentación nueva para tarea de comunicación	145
Figura 5.23. Detalle de presentación actual y presentación nueva, tarea individual simple c)	146
Figura 5.24. Detalle de presentación actual y presentación nueva, tarea individual simple d)	146
Figura 5.25. Aspecto del patrón de interacción a nivel de sistema	148
Figura 5.26. Tecnologías utilizadas en el desarrollo de la herramienta	150

Figura 5.27. Diagramas y paquetes del metamodelo de AMBUID	151
Figura 5.28. Estructura de un proyecto GMF, caso de la herramienta AMBUID	152
Figura 5.29. Fragmento del Modelo de dominio para la notación en AMBUID	153
Figura 5.30. Especificación gráfica de la Tarea de Comunicación	154
Figura 5.31. Implementación de la notación en GMF	155
Figura 5.32. Implementación de la paleta de herramientas en GMF	156
Figura 5.33. Definición del mapping para la herramienta AMBUID	157
Figura 5.34. Modelos GMF para los dos principales diagramas de AMBUID	158
Figura 5.35. Plantilla de Proceso para "Compra de Tiquetes"	161
Figura 5.36. Diagrama de Entidades de Negocio	162
Figura 5.37. Diagrama de Interfaz de Usuario de Negocio	162
Figura 5.38. Modelado de negocio para el proceso "Comprar Tiquetes de Avión"	163
Figura 5.39. Patrón de Presentación y diagrama de navegación para la tarea "Pagar Tiquete de Avión"	165
Figura 5.40. Diagrama de tareas interactivas y diagrama de diálogo	166
Figura 5.41. Mappings entre View part y view model	167
Figura 5.42. Mappings entre diagrama de máquina de estados y diagrama de dominio	167
Figura 5.43. Mappings entre diagrama de navegación y máquina de estado	168
Figura 5.44. Mappings entre diagrama de navegación y view part	168
Figura 5.45. Vista de propiedades para las acciones	169
Figura 5.46. Vista de propiedades para los eventos	169
Figura 5.47. Interfaz de usuario final para la tarea "Pagar tiquete de avión"	170

LISTA DE TABLAS

Tabla 2.1. Ejemplo de la vista de Zachman para un sistema de gestión de congresos	22
Tabla 2.2. Descripción de Taxonomías en Ingeniería de Software y HCI	55
Tabla 2.3. Características Generales de Taxonomías en Ingeniería de Software y HCI	57
Tabla 3.1. Aspectos en los que se enfocan las propuestas analizadas	68
Tabla 3.2. Clasificación taxonómica inicial-capa funcionalidad	69
Tabla 3.3. Clasificación taxonómica inicial-capa colaboración	70
Tabla 3.4. Clasificación taxonómica inicial-capa UI	71
Tabla 3.5. Clasificación taxonómica de la actividad para datos, motivación, personas, tiempo y ubicación, capa funcionalidad	81
Tabla 3.6. Clasificación taxonómica de la actividad para datos, motivación, personas, tiempo y ubicación, capa colaboración	82
Tabla 3.7. Clasificación taxonómica de la actividad para datos, motivación, personas, tiempo y ubicación, capa interfaz de usuario	82
Tabla 3.8. Clasificación taxonómica de la actividad: elementos de modelado que denotan comunicación, capa funcionalidad	90
Tabla 3.9. Clasificación taxonómica de la actividad: elementos de modelado que denotan comunicación, capa interfaz de usuario	90
Tabla 3.10. Clasificación taxonómica de la actividad: elementos de modelado que denotan comunicación, capa colaboración	91
Tabla 3.11. Proceso de clasificación de la TxA	92
Tabla 4.1. Tipo de restricción según el nivel de abstracción	102
Tabla 5.1. Descripción del flujo primario caso de uso-realizar recaudo	112
Tabla 5.2. Información del CU-Realizar Recaudo-ubicada sobre la Estructura de la TxA	115
Tabla 5.3. Listado de observaciones generales al CU-Realizar recaudo	117
Tabla 5.4-Reporte de Comentarios y Recomendaciones al Caso de Uso-Realizar Recaudo	118



INTRODUCCIÓN

CAPÍTULO PRIMERO

1. INTRODUCCIÓN

1.1 MOTIVACIÓN

1.2 OBJETIVOS

1.2.1 Objetivo general

1.2.2 Objetivo específico 1

1.2.3 Objetivo específico 2

1.2.4 Objetivo específico 3

1.2.5 Objetivo específico 4

1.2.6 Objetivo específico 5

1.3 METODOLOGÍA

1.3.1 Definición del dominio y contexto

1.3.2 Construcción del marco de referencia

1.3.3 Determinación del problema

1.3.4 Formulación de la hipótesis

1.3.5 Propuesta de investigación

1.3.6 Validación de la propuesta

1.3.7 Análisis de resultados o conclusiones

1.3.8 Resumen



CAPÍTULO PRIMERO

1. INTRODUCCIÓN

Los Sistemas de Información, así como los sistemas interactivos transaccionales son parte vital en las organizaciones modernas. Durante el desarrollo de este tipo de sistemas, se capturan grandes volúmenes de información que contienen características y atributos tanto de la organización, como del sistema interactivo. Particularmente, la parte del diseño que corresponde a la especificación de la funcionalidad en los sistemas interactivos, está comúnmente dirigida por los requisitos. Las especificaciones de dichos requisitos se basan en plantillas que clasifican la actividad de una forma que en la mayoría de los casos, los desarrolladores no son conscientes y por tanto no comprenden cómo una mala clasificación, debido a deficiencias metodológicas, afecta el entendimiento que tiene el usuario en relación con el sistema que está utilizando. Una situación ideal es que el desarrollador comprenda perfectamente cuál es el impacto de una mala clasificación y le ayude a reconocer los límites que le permitan definir un alcance para la usabilidad, la seguridad o cualquier otro aspecto.

Por otro lado, el diseño de la interfaz de usuario está mediado por muchos intereses. Por ejemplo, intereses desde la marca, el marketing, la etnografía, la accesibilidad, la presentación, la seguridad, todos ellos orquestados a través de las acciones de la interfaz. Estos intereses provienen de las contribuciones de múltiples expertos que entienden el sistema interactivo desde diferentes perspectivas y puntos de vista. El problema radica en cómo se combinan y se organizan dichas contribuciones cuando se encuentran especificadas en lenguaje natural y expresadas en diferentes tipos de notaciones.

Dentro de este escenario se sitúa la Clasificación del Modelado de la Actividad (Giraldo, 2010) que se refiere a encontrar las distintas clasificaciones de la actividad según los distintos aspectos del desarrollo de software, funcionalidad, colaboración, seguridad, desarrollo de la interfaz de usuario, etc. La clasificación del modelado de la actividad busca identificar las categorías en las que se puede estructurar toda la labor que se encuentra presente mientras se busca una meta de negocio. En este sentido, el interés de este trabajo se enfoca en comprender cómo esta información, obtenida desde la clasificación, puede ser enlazada e integrada para que contribuya a la creación de artefactos que expresen la interfaz de usuario de un sistema interactivo.

La meta que se ha perseguido en la elaboración de la presente tesis es la definición de un marco conceptual, que apoyado por un buen marco de referencia, permita la definición de metodologías, notaciones y herramientas, todo esto dentro de un



roadmap, que es el resultado de la interpretación del Marco de Desarrollo CIAF (Collaborative Interaction Application Framework) (Giraldo, 2010).

CIAF es un marco de desarrollo para sistemas interactivos colaborativos, que integra el desarrollo de la Funcionalidad, de la Colaboración y de la Interfaz de Usuario. Particularmente, para esta última, propone una metodología llamada TD-MBUID (Task & Data-Model Based User Interface Development) que se centra en combinar el diseño de las interfaces basadas en modelos de datos y de tareas.

El marco conceptual propuesto en esta tesis, adicionalmente, sirve como un marco de evaluación para propuestas metodológicas que permite identificar qué aspectos están siendo capturados en las descripciones de la labor, por ejemplo la usabilidad, la interfaz de usuario, la colaboración, la funcionalidad, la seguridad, etc. Dicho marco conceptual puede ser extendido para definir nuevos lenguajes, metodologías y herramientas que soporten el modelado y la ejecución de la actividad de sistemas interactivos.

1.1 MOTIVACIÓN

La creciente complejidad que existe a la hora de desarrollar sistemas interactivos transaccionales es la principal motivación que está detrás de esta tesis doctoral. En particular, es de interés la investigación en el dominio de la Ingeniería de Software (IS) que se centra principalmente en generar productos software completamente funcionales (Trætteberg, 2002) y en el contexto de la Interacción Humano Computador (HCI) que se centra más en capturar el modelo mental de los usuarios y en otros aspectos relacionados a la psicología cognitiva y al factor humano (Granollers, 2004). Se trata de definir soluciones que, mediante el uso de tecnologías, apoyen la transformación sistemática de descripciones que se sitúan en el *nivel del problema* para obtener implementaciones a *nivel de software* (Frankel, 2004a), tal y como lo promueve la Ingeniería Dirigida por Modelos (MDE).

Existen muchas razones por las cuales es importante construir productos software. Una de ellas es porque el software afecta muy de cerca a cualquier aspecto de nuestra vida y está muy extendido en nuestro comercio, cultura y en nuestras actividades cotidianas (Scott, et al, 2001). Usamos software para que haga cosas por nosotros, es decir, el software tiene una función primaria que es automatizar la labor de los humanos. Esta labor debe descomponerse para identificar las tareas que hace el humano, las tareas que hace el sistema y las tareas que hacen en conjunto. Esta descomposición es materia de estudio de etnógrafos, de analistas e ingenieros de software. Normalmente, todo esto se realiza pensando en el aspecto de la funcionalidad pero existen otros tipos de actividades que el software debe soportar como el hecho de que las personas colaboren, se comuniquen, intercambien información, etc. Toda esta labor debe estar automatizada también para que el sistema la pueda hacer. Esta es la razón por la que cualquier metodología que pretenda generar software funcional, debe considerar un modelado de la actividad. Es necesario entonces tener una clasificación de la actividad

para llevarla de lo tácito a lo explícito, en términos computacionales y soportar así, la automatización como función primaria del software.

Los análisis realizados a diversas propuestas metodológicas para el desarrollo de software, entre ellas, OpenUp (Balduino, 2007), SCRUM (James, 2013), CIAM (Molina et al, 2006), IDEAS (Lozano, 2001), Wisdom (Nunes, 2001), permiten observar que todas ellas se articulan a partir de la labor, es decir, como lo indica Giraldo (Giraldo, 2010), la actividad se toma casi siempre como un punto de referencia que permite interrelacionar las distintas variables que dirigen el diseño.

En este sentido, hemos entendido que la función o la labor que se realiza tanto a nivel de negocio como a nivel del sistema interactivo, es el eje que es transversal a todas las perspectivas y propuestas metodológicas para el desarrollo de sistemas interactivos. Por esta razón, hemos centrado esta investigación en entender de la manera más detallada cómo representar la labor o actividad de un sistema.

Existe un gran número de propuestas para representar la actividad en los sistemas interactivos. Sin embargo, se detecta que existen importantes deficiencias:

1. Las notaciones actuales utilizadas para describir la actividad que tiene lugar tanto en la "inter-acción" entre las personas (organización) como en la "interacción" entre el usuario y el sistema interactivo, por ejemplo: GOMS (Stuart et al, 1983), CTT (Paternò, 2004), CIAM (Molina et al, 2006), UML v.1.1 (OMG, 1999), HTA (Annett et al, 1967), HAMSTERS (Barboni et al, 2010), etc, no distinguen (no clasifican) toda la información relacionada con el modelado de la actividad asociada a distintos clasificadores, por ejemplo: nivel de la granularidad, nivel de abstracción, tipo de actividad, modelado esencial, etc. es decir, no proveen suficientes elementos de modelado, por tanto, no soportan de manera adecuada e integrada el modelado de las responsabilidades (personas "organización"), de la navegación (diálogo externo asociado a la actividad de grano grueso) y del diálogo (comunicación fina con el sistema interactivo independiente de la modalidad). De esta forma no es posible definir herramientas de modelado, de generación automática, transformación y validación que hagan que el proceso de desarrollo de la interfaz de usuario y de la funcionalidad se haga de manera sistemática con apoyo de metodologías y herramientas consistentes.
2. Dentro de las notaciones (mencionadas anteriormente) más difundidas se observa una gran cantidad de adaptaciones a su sintaxis concreta que se originan por la necesidad de agregar información, a los modelos, relacionada con otros aspectos distintos al modelado de la actividad. Esto genera una reducción en la expresividad de los diagramas resultantes. Frecuentemente, esto sucede con notaciones muy reconocidas que cuentan con un alto nivel de soporte por medio de herramientas para la edición y automatización. Por ejemplo, CTT (Diagrama interacción) (Paternó, 2004) se ha adaptado ligeramente para explotar (rentabilizar) al máximo dicha notación pero dando como resultado diagramas híbridos (p.e. los desarrolladores utilizan CTT para representar la interfaz de usuario).

3. La necesidad de incorporar un mayor número de notaciones para aumentar la expresividad y la precisión de la representación del sistema interactivo genera una gran carga cognitiva para los desarrolladores cuando intentan entender, la ejecución de los modelos del sistema sin contar con una buena representación del mismo en tiempo ejecución (runtime) o unas buenas herramientas que faciliten la implementación de dichos modelos. Así mismo, como no está definido cuál sería una forma estándar de representación del modelo de implementación (Low Level Specification) de un sistema interactivo, los desarrolladores tampoco reciben apoyo en el proceso de transformación de los modelos de diseño al modelo de implementación.
4. Además, los desarrolladores (codificadores) que se enfrentan al proceso de traducir los modelos (diferentes diagramas) al respectivo código de la aplicación, de la interfaz, la navegación y el diálogo no encuentran el suficiente apoyo de herramientas de tiempo de ejecución (middleware) y máquinas virtuales que provean servicios para la ejecución de los distintos modelos del diseño traducidos.

Estas limitaciones justifican la falta de un marco conceptual, metodológico y tecnológico, centrado en el Modelado de la Actividad, que permita la definición de notaciones, herramientas y metodologías que soporten el desarrollo de sistemas interactivos. Con el objetivo de cubrir las necesidades y limitaciones comentadas, se plantea la investigación que ha dado lugar a esta tesis. Este trabajo de investigación parte de la formulación de la pregunta de investigación:

*¿Cómo extender los lenguajes actuales utilizados para describir la actividad que tiene lugar tanto en la "inter-acción" entre las personas (organización) como en la "interacción" entre el usuario y el sistema interactivo para incrementar su expresividad y mejorar el **modelado de la actividad** e incorporar la generación de modelos de ejecución que sean soportados por una herramienta runtime, todo esto dentro del contexto de la metodología CIAF en todos sus componentes = CIAFHCI (TD-MBUID), CIAFFUN y CIAFCSCW?*

Dicha pregunta de investigación conduce a la formulación de la siguiente hipótesis:

*"Es factible la definición de un **marco conceptual, metodológico y tecnológico** basado en la Clasificación de la Actividad, que contribuya a mejorar el modelado de la actividad que tiene lugar tanto en la "inter-acción" entre las personas (organización) como en la "interacción" entre el usuario y el sistema interactivo, mejorando la expresividad de los lenguajes actuales utilizados para describir la actividad e incorporando la generación de modelos de ejecución que sean soportados por una herramienta runtime, todo esto dentro del contexto de la metodología TD-MBUID."*

Esta hipótesis general supone la validación de dos sub-hipótesis. Dichas hipótesis secundarias quedan formuladas en los siguientes términos:

*H1: "Es factible la definición de un **marco conceptual** basado en la Clasificación de la Actividad, que contribuya a mejorar el modelado de la actividad que tiene lugar tanto en la "inter-acción" entre las personas (organización) como en la "interacción" entre el usuario y el sistema interactivo, mejorando la expresividad de los lenguajes actuales utilizados para describir la actividad, todo esto dentro del contexto de la **metodología TD-MBUID**."*

*H2: "Es factible la definición de un **marco tecnológico** basado en la Clasificación de la Actividad, que incorpore la generación de modelos de ejecución que sean soportados por una herramienta runtime, todo esto dentro del contexto de la **metodología TD-MBUID**."*

1.2 OBJETIVOS

Las hipótesis antes enunciadas implican la definición de una serie de objetivos y sub-objetivos a cumplir en el contexto de la presente tesis.

1.2.1 Objetivo general:

El objetivo general de esta investigación es el de Re-diseñar y extender el Marco de Desarrollo CIAF para extender los lenguajes actuales aumentando su expresividad e incorporar la generación de modelos de ejecución que sean soportados por una herramienta *runtime*, todo esto dentro del contexto de la metodología TD-MBUID.

1.2.2 Objetivo específico 1:

Especificar un *Marco Conceptual*, que extienda el marco conceptual de CIAF (dentro del contexto de TD-MBUID), que permita integrar los lenguajes actuales más utilizados para describir el modelado de la actividad y que soporte de manera integrada el modelado de las responsabilidades, navegación y el diálogo entre ellos.

Estrategia 1: Realizar una revisión sistemática de las propuestas existentes para modelar la actividad en sistemas interactivos.

Estrategia 2: Evaluar las capacidades de cada propuesta, sus puntos débiles, sus ventajas y su potencialidad para la integración.

Estrategia 3: Llevar las propuestas a un mismo nivel de formalización que haga posible la identificación de puntos de integración entre ellas.

Estrategia 4: Evaluar los distintos puntos donde es posible explotar la complementariedad de los procesos en torno a la idea de que se busca crear.



Estrategia 5: Llevar a cabo una definición de los contenidos finales que resultan de la unión de las propuestas.

1.2.3 Objetivo específico 2:

Adaptar la *sintaxis concreta* de cada lenguaje utilizado para describir el sistema interactivo (soportados por el marco conceptual extendido), de acuerdo a los intereses de cada rol y al propósito para el que está diseñado con el fin de aumentar su nivel de expresividad.

Estrategia 6: Analizar la sintaxis concreta de cada lenguaje utilizado para describir el sistema interactivo e integrado con el marco conceptual CIAF.

Estrategia 7: Evaluar el nivel de expresividad de los elementos que conforman los lenguajes de las propuestas integradas con CIAF.

Estrategia 8: Modificar la sintaxis concreta para los elementos que conforman los lenguajes de las propuestas integradas con CIAF, según los resultados de la evaluación.

Estrategia 9: Definir nuevos elementos en la sintaxis concreta para la descripción del sistema interactivo en el Marco CIAF, según los resultados de la evaluación.

1.2.4 Objetivo específico 3:

Extender el *marco tecnológico* de CIAF para que soporte la edición de los modelos propuestos y la generación de los nuevos modelos de ejecución y modelos de implementación a partir de los diferentes modelos de diseño de sistemas interactivos.

Estrategia 10: Analizar el funcionamiento y composición del marco tecnológico de CIAF en cuanto a la edición de los modelos propuestos.

Estrategia 11: Especificar los componentes a integrar en CIAF, necesarios para que soporte la edición de los modelos propuestos.

Estrategia 12: Implementar los componentes especificados.

Estrategia 13: Integrar los componentes implementados al marco tecnológico de CIAF.

1.2.5 Objetivo específico 4:

Implementar un *conjunto de librerías de runtime* que provean los servicios requeridos para la ejecución de los modelos de ejecución e implementación generados por la herramienta.

Estrategia 14: Especificar las librerías de *runtime* necesarias para que provean los servicios requeridos para la ejecución de los modelos de ejecución e implementación generados por la herramienta.

Estrategia 15: Implementar las librerías *runtime*.

Estrategia 16: Integrar las librerías *runtime* al marco tecnológico de CIAF.

1.2.6 Objetivo específico 5:

Validar la propuesta mediante la aplicación de un caso de estudio.

Estrategia 17: Definir el caso de estudio para validar el marco conceptual, metodológico y tecnológico propuesto.

Estrategia 18: Validar el marco propuesto aplicándolo al caso de estudio.

Estrategia 19: Concluir resultados de la validación con el caso de estudio.

1.3 METODOLOGÍA

Para validar las hipótesis y alcanzar los objetivos que se plantean en esta tesis se propone combinar los métodos de Investigación Básica e Investigación Aplicada, tal y como lo hace Giraldo en su tesis (Giraldo, 2010). Ambos métodos garantizan la consecución de excelentes resultados para objetivos relacionados con el desarrollo de marcos teóricos y con el diseño e implementación de sistemas aplicados a las áreas de Ingeniería de Software e Interacción Humano Computador.

En particular la *Investigación Básica* se aplica cuando se han identificado problemas para los que se persigue el desarrollo de métodos sistemáticos que puedan ayudar a resolverlos (Tamayo, 1999). Por otro lado, la *Investigación Aplicada* permite validar métodos, técnicas y sistemas para los que se exigen ciertos requerimientos funcionales, de soporte a la interacción o de satisfacción de los usuarios. Así, este método se caracteriza por su orientación en la obtención de resultados en los productos que se construyen y que serán evaluados y contrastados con los objetivos iniciales (Tamayo, 1999). Por lo tanto, se propone seguir, para el desarrollo de esta investigación, una combinación de instancias de los métodos anteriormente citados.

Además, en el curso de esta investigación se van a desarrollar diversos productos software (herramientas de modelado, editores, software de soporte a transformaciones, etc.). Para el desarrollo de dicho software se seguirá el proceso RUP (*Rational Unified Process*) enmarcado en un ciclo de vida iterativo e incremental. Así, se irán construyendo primeras versiones que pueden ser sometidas a pruebas de funcionalidad y de usabilidad desde las fases más tempranas del desarrollo.

En la Figura 1.1 se ilustra de manera resumida la concepción de la tesis de investigación, dentro de la cual se muestran los principales componentes que fundamentan el análisis de las problemáticas que son abordadas antes de identificar un posible conjunto de soluciones. Una investigación organizada, crítica, sistemática, científica y basada en datos permite encontrar respuestas o soluciones a determinados problemas (Sekaran, 2006). El propósito es identificar cuáles son los *objetos de estudio*

que tienen relevancia dentro del marco de desarrollo. Se entiende por objeto de estudio cualquier componente metodológico que pertenezca a la solución. Los objetos de estudio son identificados para entender su estado actual, y de esta forma, entender sus situaciones problemáticas y sus estados ideales. De esta forma, es posible determinar el trabajo necesario para acercar cada objeto de estudio a su estado ideal. Una vez que se han resuelto las situaciones problemáticas de cada objeto de estudio se puede dar por finalizada la investigación.

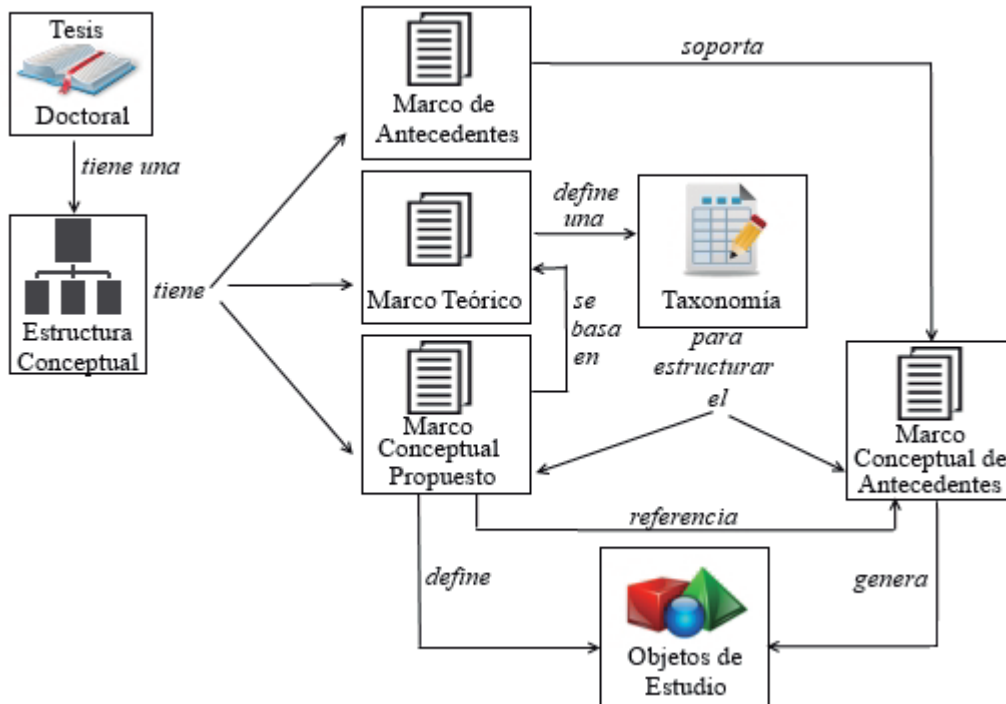


Figura 1.1. Concepción del trabajo de investigación (Giraldo, 2010).

1.3.1 Definición del dominio y contexto

En la etapa más temprana, antes de iniciar el esfuerzo de investigación, es necesario definir el *dominio* y el *contexto de la investigación*. Esta tesis se enmarca principalmente dentro del dominio de la Ingeniería de Software. Es por esto que, básicamente para todos los contextos en los que se desenvuelve la tesis, es la Ingeniería de Software el eje director que influencia la ejecución de ciertas etapas mínimas requeridas (análisis, diseño, implementación, pruebas, etc.). Obviamente, estas etapas consideran técnicas y artefactos muy propios de su disciplina para cada uno de los contextos. Los principales contextos a tener en cuenta son: los sistemas interactivos (HCI), modelado de la actividad, ingeniería dirigida por modelados (MDE), entre otros. De esta manera se delimita el espectro de la investigación y se definen una serie de conceptos, a modo de palabras clave, que facilitarán los procesos de búsqueda de información.

1.3.2 Construcción del marco de referencia

Una vez definida la frontera de investigación se construye el *marco de referencia* que posee la información conceptual en la que se fundamentará el desarrollo de la tesis. En esta actividad se realizan una serie de revisiones sistemáticas (Kitchenham, 2004) que permiten identificar las propuestas más relevantes en el campo en el que esta investigación se sitúa. A partir de las revisiones sistemáticas es posible generar el *marco de antecedentes* (estado del arte) y el *marco teórico*. El marco teórico es el resultado de una investigación preliminar sobre teorías, consolidadas en la literatura, que se está siguiendo como modelo de la realidad y, que se utiliza para tomar decisiones en el diseño y para ordenar y estructurar la investigación. En esta tesis se obtiene una taxonomía, a partir del marco teórico, para organizar las propuestas que provienen del marco de antecedentes. El *marco de antecedentes* es un conjunto de antecedentes que sirven como referencia y que se encuentran en publicaciones o investigaciones recientes. Este marco orienta la investigación y es útil para identificar las ventajas y desventajas de las propuestas existentes, de tal forma que, no se cometan de nuevo los mismos errores y sea posible explotar al máximo los elementos más útiles en el contexto de esta tesis.

La complejidad que existe en el proceso de identificación y selección de objetos de estudio es muy alta debido principalmente a la diversidad de propuestas y a su alta heterogeneidad. El manejo de múltiples marcos conceptuales hace que sea difícil reconocer cuando un conjunto de conceptos que provienen de distintas propuestas están relacionados o son equivalentes. El uso de una taxonomía para tal fin permite ubicar e identificar los conceptos más relevantes de cada propuesta y determinar qué categorías carecen por completo de aportación hasta el momento.

De esta manera se obtienen tanto el marco conceptual de antecedentes como el marco conceptual propuesto. El marco conceptual de antecedentes es, por tanto, el resultado de la clasificación, por medio de la taxonomía, de todos los marcos conceptuales de las propuestas estudiadas. El marco conceptual propuesto es el resultado de extender y refinar el marco conceptual de antecedentes, para que después de un proceso continuado de desarrollo y pruebas pueda convertirse en el marco conceptual de la tesis.

1.3.3 Determinación del problema

A partir de la definición del marco conceptual de antecedentes se generan los objetos de estudio que son de interés en la investigación. Una vez generados los objetos de estudio se identifican sus condiciones actuales, sus situaciones problemáticas y sus situaciones deseadas. A partir de un proceso de contraste de estas situaciones surge la problemática a la cual se enfrenta esta investigación.

Las situaciones deseadas que se plantean para todos y cada uno de los objetos de estudio, se convertirán en las metas que representan el aporte científico que subyace en esta investigación en relación al conjunto de problemáticas identificadas.



1.3.4 Formulación de la hipótesis

Una vez determinado el problema de investigación se hace el *planteamiento de las hipótesis de investigación*. A partir de la hipótesis de investigación surge la especificación de los objetivos y las estrategias a seguir con el fin de elaborar una propuesta de investigación.

1.3.5 Propuesta de investigación

La *propuesta de investigación* tiene como objetivo principal conducir el ciclo de vida de los objetos de estudio desde su estado actual hasta el estado deseado. Dicha propuesta es modelada formalmente para que permita ser definida de manera computacional. Una vez implementada la propuesta debe estar capacitada para validar la hipótesis inicial de este proceso de investigación.

1.3.6 Validación de la propuesta

La propuesta de solución planteada debe ser verificada mediante su aplicación a casos de estudio concretos que exploren todas sus capacidades y permitan establecerla como válida (Yin, 2003).

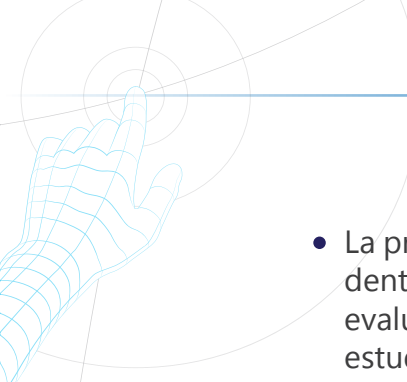
1.3.7 Análisis de resultados o conclusiones

Una vez finalizado el proceso antes mencionado se determina el nivel de cumplimiento de los objetivos inicialmente planteados, así como la importancia y el alcance de la propuesta formulada.

1.3.8 Resumen

Para construir el marco conceptual, metodológico y tecnológico objeto de esta tesis se definen y ejecutan una serie de tareas, a saber:

- La primera tarea que se propone consiste en evaluar las principales propuestas dentro de cada disciplina con el propósito de identificar sus capacidades, evaluar su adaptación a MDE y su posibilidad de integración. Las propuestas estudiadas corresponden a componentes que puedan pertenecer a un marco de desarrollo, principalmente notaciones, marcos conceptuales, componentes metodológicos, procesos, y métodos.
- La segunda tarea consiste en clasificar las propuestas seleccionadas a partir de una taxonomía y a partir de un método de clasificación del modelado de la actividad. Dicha clasificación se hace con el objetivo de identificar sus aportes, sus deficiencias y las posibilidades de integración entre propuestas.
- La tercera tarea consiste en escoger unas propuestas de interés que, individualmente y en conjunto, satisfagan una serie de requisitos necesarios para la definición del marco conceptual y tecnológico.

- 
- La primera tarea que se propone consiste en evaluar las principales propuestas dentro de cada disciplina con el propósito de identificar sus capacidades, evaluar su adaptación a MDE y su posibilidad de integración. Las propuestas estudiadas corresponden a componentes que puedan pertenecer a un marco de desarrollo, principalmente notaciones, marcos conceptuales, componentes metodológicos, procesos, y métodos.
 - La segunda tarea consiste en clasificar las propuestas seleccionadas a partir de una taxonomía y a partir de un método de clasificación del modelado de la actividad. Dicha clasificación se hace con el objetivo de identificar sus aportes, sus deficiencias y las posibilidades de integración entre propuestas.
 - La tercera tarea consiste en escoger unas propuestas de interés que, individualmente y en conjunto, satisfagan una serie de requisitos necesarios para la definición del marco conceptual y tecnológico.
 - La cuarta tarea consiste en adecuar cada propuesta seleccionada para garantizar la coherencia del conjunto sin que pierdan sus propiedades iniciales.
 - La quinta tarea consiste en complementar las propuestas, y de esta forma, dar fin a la construcción del marco conceptual y tecnológico.

En este capítulo se presenta la conceptualización del desarrollo de la tesis. Esta conceptualización propone unas hipótesis de investigación y al mismo tiempo una metodología a seguir. En toda investigación una vez que se hace esta conceptualización, es importante hacer una revisión exhaustiva de los trabajos existente en el área de interés. En el siguiente capítulo se presenta el marco de referencia de esta investigación.





2

MARCO DE REFERENCIA

CAPÍTULO SEGUNDO

2. MARCO DE REFERENCIA

2.1 FUNDAMENTOS TEÓRICOS

2.1.1 Ingeniería de software

2.1.2 Casos de uso

2.1.3 Workflow

2.1.4 Teoría de la actividad

2.1.5 Ingeniería semiótica

2.1.6 Análisis de la comunicación

2.1.7 Interacción humano computador (HCI)

2.2 ANTECEDENTES

2.2.1 Modelado y ejecución de la actividad en sistemas interactivos, funcionalidad

2.2.2 Modelado y ejecución de la actividad en sistemas interactivos, colaboración

2.2.3 Modelado y ejecución de la actividad en sistemas interactivos, interfaz de usuario

2.2.4 Taxonomías en ingeniería de software y HCI



CAPÍTULO SEGUNDO

2. MARCO DE REFERENCIA

En este capítulo se describe el marco de referencia en el que se fundamenta el desarrollo de esta investigación. En una investigación nunca se parte de cero, sino que se parte de una base teórica y conceptual determinada. Ella guía todo el proceso, y en base a ella llegamos nuevamente al objetivo de toda investigación: generar un conocimiento válido y generalizable (Tamayo, 1999).

El marco de referencia que se presenta en esta tesis consta de dos apartados principales: el marco teórico, y los antecedentes. El marco teórico es el resultado de una investigación preliminar sobre teorías, consolidadas en la literatura, que se están siguiendo como modelo de la realidad y, que se utiliza para tomar decisiones en el diseño y para ordenar y estructurar la investigación. El marco de antecedentes es un conjunto de antecedentes que sirven como referencia y que se encuentran en publicaciones o investigaciones recientes. Finalmente, se presentan las conclusiones en las que se analizan, por medio de clasificaciones, las propuestas estudiadas.

2.1 FUNDAMENTOS TEÓRICOS

En esta sección se presentan algunas descripciones teóricas sobre temas que son relevantes en el desarrollo de esta investigación. Estos temas han sido estudiados como parte fundamental de la formación de la comunidad de ingeniería de software. En este sentido, se considera que son pilares para establecer la coherencia y consistencia del marco conceptual y tecnológico. Los fundamentos teóricos proporcionan, principalmente, la terminología que es utilizada para definir la propuesta de investigación.

2.1.1 Ingeniería de Software

La ingeniería de software es el área central de esta investigación, considerada como el dominio que enmarca y delimita el interés de investigación. Desde este punto de vista, se puede decir que los temas tratados en esta tesis tienen una relación directa, o un énfasis, con la ingeniería de software. Otras áreas transversales, como la interacción humano computador, la interfaz de usuario, son consideradas áreas en las cuales se desarrolla software en distintos contextos o enfoques. A continuación, se abordan los temas principales, tanto a nivel de dominio como a nivel de contexto.



2.1.1.1 Taxonomías

El concepto "*Taxonomía*", tiene sus raíces en el griego "*Taxis* = ordenación" y "*Nomos* = reglas" y según el diccionario de la RAE (RAE, 2015) se define como "*Ciencia que trata de los principios, métodos y fines de la clasificación*". Normalmente, los trabajos relacionados con la definición de Taxonomías en Ingeniería de Software y en HCI, no hacen referencia a esta definición, aunque como lo reconoce (Lessiter et al, 2014) sí se menciona que el término fue inicialmente utilizado en las ciencias biológicas y que se ha ido expandiendo a cualquier "clasificación ordenada".

Por otro lado, la clasificación es una actividad inherente a los humanos, pero cuando se habla de "*Taxonomía*" puede resultar un poco intimidante y complejo. Puede ser por el hecho de que definir y utilizar una taxonomía suele ser más exhaustivo que el hecho de definir y utilizar una clasificación o una estructura (Oxton et al, 2015).

Las taxonomías se han utilizado con frecuencia en la gestión de documentos y, hoy día, sobre todo en la Web (Cleophas et al, 2005), (Zhang y Lee, 2004), (Tzitzikas et al, 2005), (Spangler et al, 2003). Una taxonomía puede ser una simple organización de clases de cosas en grupos. Las taxonomías son consideradas como menos amplias que las ontologías debido a que las ontologías aplican una mayor variedad de tipos de relación. Es decir, las taxonomías permiten básicamente la categorización de conceptos mediante unos métodos y reglas que definen la estructura de una ontología.

Las taxonomías son probablemente la herramienta más antigua y más ampliamente usada para modelado conceptual, no obstante, es una potente herramienta empleada en la construcción de marcos conceptuales. De ahí que sea habitual encontrar estructuras taxonómicas (comúnmente llamadas taxonomías) en las propuestas. En esta tesis se hace una distinción entre la *estructura taxonómica* (matriz clasificatoria) y el método para llevar a cabo la clasificación que tiene lugar para obtener dicha estructura taxonómica. Es decir, una *taxonomía* define tanto el método de clasificación como la estructura taxonómica.

2.1.1.2 Taxonomías en Ingeniería de Software

La Ingeniería de Software, como cualquier otra disciplina requiere de herramientas que permitan estructurar y comunicar el conocimiento, así como avanzar en su entendimiento. Por tal razón, el desarrollo de taxonomías es esencial para documentar teorías que acumulan conocimiento en dicha disciplina (Unterkalmsteiner et al, 2014).

La guía sobre el cuerpo de conocimiento de la Ingeniería de Software (SWEBOK) (Abran et al, 2004) puede ser clasificada como una taxonomía, ya que caracteriza la disciplina de la Ingeniería de Software y provee un acceso estructurado a su cuerpo de conocimiento y promueve la comunicación estructurada de esta disciplina. Existen varias taxonomías en el campo de investigación de la Ingeniería de Software, que serán nombradas en el marco de antecedentes. Particularmente, para este trabajo interesa profundizar en la descripción de una de las estructuras taxonómicas más representativas

a nivel de clasificación de elementos que conforman sistemas empresariales.

A continuación se describe el Framework de Zachman; uno de los framework más representativos a nivel de descripción de sistemas empresariales. Este framework es utilizado en esta tesis como uno de los pilares conceptuales más importantes, tanto para definir taxonomías como para hacer las clasificaciones.

2.1.1.2.1 Framework de Zachman

Zachman (Zachman, 1987) introduce su marco de trabajo en el año 1987 como una respuesta al incremento del tamaño y la complejidad de los sistemas de información. En su propuesta define una arquitectura de sistemas de información para crear un *framework* descriptivo a partir de diversas disciplinas en este campo. La primera versión del *framework de Zachman* cuenta con tres vistas: de datos, de procesos y de red. Además, no propone ninguna notación para modelar cada una de sus celdas. Sowa y Zachman (Sowa y Zachman, 1992) presentan en 1992 una extensión en la cual formalizan el *framework* y agregan las vistas de organización, tiempo y motivación.

El *framework* provee una *taxonomía* sistemática de conceptos para relacionar cosas que describen el mundo real con los conceptos que describen su respectivo sistema de información y su implementación (Sowa et al, 1992). El propósito del *framework* de Zachman es mostrar cómo las distintas aportaciones existentes para el diseño de sistemas de información se integran unas con otras. Para ello el framework incluye relaciones entre las celdas de las distintas columnas y filas que lo forman.

El *framework* se compone de treinta cajas o celdas, organizado en seis columnas y cinco filas. Para cada una de esas celdas es posible definir una notación especial que pueda describir la información de dicha celda. Pero para relacionar la información de todas las celdas debe existir un lenguaje común que pueda describir todas las celdas y sus interrelaciones.

Cada fila o **perspectiva** es una representación arquitectónica fundamental asociada a cada uno de los conjuntos de roles (planificador, propietario, diseñador, desarrollador e integrador) que está asociada a un nivel de abstracción. A continuación se describen las perspectivas del *framework* (Este framework se explica usando como metáfora la construcción de edificios) (Giraldo, 2010):

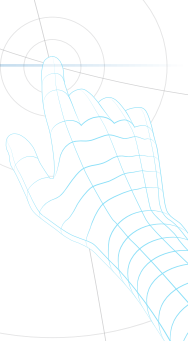
- **Alcance (Scope):** Es el primer bosquejo arquitectural del sistema, el cual ilustra, a grandes rasgos, el tamaño, la forma, la relación espacial, la infraestructura organizacional, la logística y el propósito básico de la estructura final.
- **Modelo de negocio o de empresa (Enterprise or business model):** A partir de los bosquejos del arquitecto, que representan la forma final del sistema desde la perspectiva del propietario, se producen los modelos de la empresa, los cuales constituyen el diseño del negocio.
- **Modelo de sistema (System model):** Los planes del arquitecto son la traducción

de los bosquejos a especificaciones detalladas desde la perspectiva del diseñador.

- **Modelo de la Tecnología (*Technology model*):** El contratista debe redibujar los planos del arquitecto para representar la perspectiva del constructor, los cuales deben considerar las restricciones de las herramientas, la tecnología, y los materiales.
- **Componentes (*Components*):** Los subcontratistas trabajan a partir de los planes de compra, que especifican de forma detallada las partes y subsecciones. Esto corresponde a las especificaciones detalladas que son dadas a los programadores, quienes codifican módulos individuales sin preocuparse por el contexto o estructura completa del sistema.

El *framework* plantea la existencia de diferentes tipos de descripciones relativas a diferentes aspectos de los sistemas que se describen. Cada una de las diferentes descripciones se prepara con un motivo distinto, cada una es independiente y diferente de las otras, aunque todas puedan pertenecer al mismo objeto, razón por la cual están intrínsecamente relacionadas unas con otras. Esta última idea queda plasmada gracias a la especificación de distintas vistas o abstracciones. A continuación se presentan las vistas que propone el *framework*:

- **Los Datos (qué), *The Data (What)*:** Modelan los conceptos o cosas que son importantes para el negocio y por tanto que el negocio manipula. El modelo descriptivo para especificar datos está basado en "*entidad-relación-entidad*", y el significado de entidad y relación cambia de una perspectiva a otra.
- **Los procesos (cómo), *The Process (How)*:** El modelo descriptivo para especificar procesos está basado en el esquema "*entrada-proceso-salida*".
- **La red (dónde), *The Network (Where)*:** El modelo para describir la arquitectura de la red utiliza la representación "*nodo-enlace-nodo*". Un nodo puede ser una localidad, un componente, un computador, etc.
- **Las personas (quién), *The People (Who)*:** Es útil abstraer el concepto de personas que forman la empresa debido a la importancia de diseñar la infraestructura de la organización o las relaciones existente entre los miembros de la misma. El modelo descriptivo que permite especificar las personas utiliza la representación "*gente-trabajo-gente*".
- **El tiempo (cuándo), *The Time (When)*:** El tiempo se abstrae del mundo real para diseñar las relaciones "*evento-a-evento*", que establecen los calendarios del sistema.
- **La motivación (Por qué), *The Motivation (Why)*:** Supone la representación descriptiva de la motivación de la empresa. Su modelo básico debe ser del tipo "*finés-medio-finés*", donde los fines son objetivos (o metas) y los medios son estrategias (o métodos).



El *framework* de Zachman especifica una serie de reglas a satisfacer para garantizar la consistencia de la información de los distintos modelos que representan las arquitecturas empresariales. A continuación se enumeran dichas reglas:

- **Regla 1:** Las columnas no tienen orden. Todas son igualmente importantes.
- **Regla 2:** Cada columna tiene un modelo básico simple. El modelo básico de cada columna es un metamodelo genérico. Es genérico porque es el mismo para cada celda en la misma columna (Zachman, 1987).
- **Regla 3:** El modelo básico de cada columna debe ser único. La unicidad es esencial para cualquier esquema de clasificación útil.
- **Regla 4:** Cada fila representa una perspectiva distinta y única. Cada una trata con distintas restricciones del modelo (Zachman, 1987).
- **Regla 5:** Cada celda es única (porque tiene un único modelo básico) que hace a cada columna única y pertenece a una sola perspectiva.
- **Regla 6:** La composición o integración de todos los modelos de las celdas en una fila constituye un modelo completo desde la perspectiva de esa fila.
- **Regla 7:** La lógica es recursiva.

En la Tabla 2.1 se presenta un ejemplo de clasificación de la información de un sistema de gestión de congresos (Giraldo, 2010). Puede observarse cómo se han rellenado todas las celdas del *framework* de Zachman con información que surge desde el análisis de negocio y que corresponde a la especificación de las necesidades de automatización que tiene el cliente. No toda la información que describe un sistema es necesaria para su incorporación en el *framework*. Solo es útil la información que se requiere que se transfiera al nivel de la tecnología.

2.1.1.3 Ingeniería Dirigida por Modelos (MDE)

El término *Model-Driven Engineering* (MDE) se usa comúnmente para describir los enfoques de desarrollo de software, en los cuales se crean modelos abstractos de sistemas software, que son transformados sistemáticamente en modelos más concretos y eventualmente en implementaciones concretas (France et al, 2007). Tanto el MDE, el MDSD (*Model Driven Software Development*) y el MDA (*Model Driven Architecture*) se centran en las especificaciones, ya sea de tipo abstracto, concreto, o en el meta nivel en lugar de centrarse en cómo generar el código ejecutable. La generación de código se hace de manera más o menos automática, por medio de transformaciones o traslaciones de modelos (Olivier et al, 2004). En MDE un modelo de un sistema complejo consiste en la creación de un conjunto de modelos del mismo, usando una amplia variedad de puntos de vista.

En la actualidad, MDE se centra principalmente en el desarrollo de técnicas, métodos, procesos y herramientas de soporte que reduzcan efectivamente la brecha existente

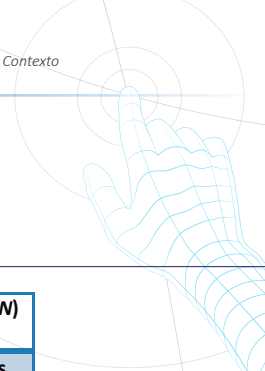
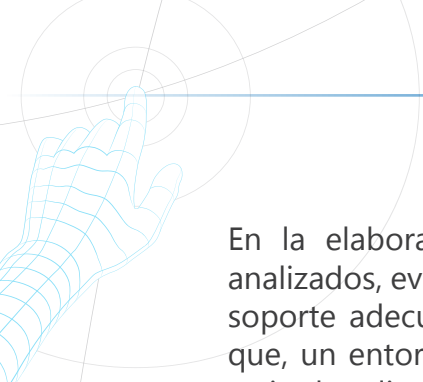


Tabla 2.1. Ejemplo de la vista de Zachman para un sistema de gestión de congresos (Giraldo, 2010).

	WHAT (DATA)	HOW (FUNCTION)	WHERE (NETWORK)	WHO (PEOPLE)	WHEN (TIME)	WHY (MOTIVATION)
SCOPE, {Contextual}, Planner.	Congreso, Artículo, Tema, Autor	Crear congreso, Evaluar artículo, Definir fechas	Ciudad Real, Madrid, Bogotá, Londres	PresidenteCP, MiembroCP, Revisor, Autor	Fecha envío, Recep, Notifica, periodo evaluación	100% de artículos recibidos, evaluados y seleccionados, disponibilidad 7/24
BUSINESS MODEL, {Conceptual}, Owner.	Autor, Congreso, listado de Artículos Aceptados, Asunto, Revisión	{Registrar, Ingresar, Evaluar} Artículo	Ciudad Real, Madrid, Bogotá, Londres	MiembroCP puede delegar en Revisor, Co- Autor puede ser Autor	Camera Ready, Fecha de envío, Recep, Notificación, periodo de evaluación	Recibirán papers hasta la fecha de submit. Se inscribirá al menos un autor. Un revisor puede ser autor. Los papers deben estar en la plantilla del congreso.
SYSTEM MODEL, {logical}, Designer	Autor, Lista final = Artículo-- Revisión. Congreso, AutorPaper	Submit Paper, Enviar correo electrónico. Actualizar fichero. Almacenar Versión	Servidor UCLM, repositorio de datos, aplicaciones cliente	Admin administra el sistema, Autor tiene interfaces de acceso	Si fecha límite -> enviar artículos a revisores	Actualizar fichero solo en las fechas previstas. Creación de password solo en las fechas. El FTP debe estar configurado
TECHNOLOGY MODEL, {physical}, Builder	Artículo: id, fichero, Plantilla, servidor	SendEmail(), UpdatePaper(), SubmitPaper()	CongressServer , Http, Iexplorer, Oracle	WebForms	Timer = date	If deadline close submit
DETAILED REPRESENTATIONS	--	--	--	--	--	--

entre el problema y la implementación (Frankel, 2004a). Uno de los aspectos que dificulta el desarrollo de software complejo es la amplia separación conceptual existente entre el dominio de discurso del problema y el de la implementación (France y Rumpe, 2007). Aunque existe una gran cantidad de tecnologías útiles para deshacer esta separación, no es fácil identificar las tecnologías más adecuadas para desarrollar software complejo sin incurrir en esfuerzos innecesarios. La complejidad de cerrar la brecha se aborda mediante el uso de modelos que describen sistemas complejos en múltiples niveles de abstracción y desde una variedad de perspectivas, así como, a través del apoyo automatizado para la transformación y el análisis de dichos modelos. De esta forma, una vez realizado el diseño, los desarrolladores utilizan tecnologías informáticas para transformar dichos modelos en sistemas completamente funcionales (France y Rumpe, 2007).

En MDE, los lenguajes de modelado deben tener definida formalmente su semántica si se desea que sean usados para crear modelos analizables (Frankel, 2004b). Por consiguiente, el trabajo en técnicas de especificación formal (FST) ha sido particularmente relevante en MDE. Esta formalización facilita la manipulación de modelos, mediante herramientas, dirigida por su semántica, ya que existe frecuentemente la necesidad de analizar formalmente un subconjunto de vistas en un modelo MDE (Favre, 2005).



En la elaboración de un proyecto software se requiere que los modelos sean analizados, evolucionados y transformados. Los entornos MDE deben proporcionar un soporte adecuado para el almacenamiento y manipulación de modelos. Es por eso que, un entorno de desarrollo de este tipo, debe proporcionar al desarrollador una serie de editores que soporten el desarrollo de grupos de modelos, así como de sus respectivas transformaciones. Además, se deberá permitir la manipulación de modelos y mantener una relación de trazabilidad entre ellos.

2.1.1.3.1 Desarrollo de Software Dirigido por Modelos (MDSO)

El desarrollo de software dirigido por modelos (MDSO), es la visión que tiene MDE dentro del campo de la Ingeniería del Software. El MDSO cubre aspectos propios de la Ingeniería del Software mediante la especificación de técnicas, métodos, herramientas y lenguajes. Cuando en un proyecto de desarrollo de software se dice que está incorporado el MDSO es porque existe una separación conceptual entre el espacio de modelado y el espacio de implementación.

Los métodos de desarrollo de software basado en modelos tienen como objetivo apoyar a los ingenieros de software en la producción automática de sistemas de software de gran tamaño, que sean muy flexibles, portables y de un alto valor para sus clientes. Fundamentalmente se pretende liberar a los programadores de una gran cantidad de tareas estándar y tediosas, que son también una importante fuente de errores. Se supone que, al aplicar sistemáticamente MDSO, la calidad de los sistemas de software, el grado de reutilización y, por tanto, implícitamente, la eficiencia de desarrollo, va a mejorar.

La idea fundamental de MDSO es que los modelos se conviertan en una especie de "código fuente" del sistema a partir de los cuales los ejecutables, simplemente, se generan. Así, los modelos cubren diferentes niveles de abstracción, que van desde esquemas conceptuales en el espacio del problema a los modelos detallados de bajo nivel adaptados a una plataforma específica. En general, el MDSO es el proceso de generación de sistemas de software ejecutable a partir de modelos formales. Este proceso comienza a partir de modelos independientes de la computación (CIM), que se transforman en modelos independientes de la plataforma (PIM) para adaptarse a los modelos de plataforma específica (PSM) y, finalmente, dar como resultado el código fuente final de la aplicación (por ejemplo, Java) (Rech et al, 2009). Una de las principales iniciativas en MDSO es MDA de la OMG (Object Management Group).

2.1.1.3.2 Lenguajes específicos de dominio (DSLs) y lenguajes ejecutables

En su trabajo "*MDE Basis with a DSL Focus*", Andova (Andova et al, 2012) define la palabra lenguaje como una colección de sentencias o expresiones, construidas de acuerdo a ciertas reglas gramaticales, donde los elementos del lenguaje se refieren a entidades del mundo real. En este sentido, en el mismo trabajo definen un DSL como un lenguaje formal, que se puede procesar, dirigido a un aspecto específico de una



tarea o de un sistema de procesamiento de información, como por ejemplo construcción de interfaces de usuario, realización de consultas a una base de datos, construcción de páginas web, intercambio de datos, generación de escáneres y analizadores. No existe un requerimiento acerca de que un DSL deba estar enteramente completo, a diferencia de un lenguaje de propósito general. Su semántica, flexibilidad y notación están diseñadas para soportar el trabajo con dichos aspectos tan eficientemente como sea posible.

Por su parte, Selic (Selic, 2012) define un conjunto de características en el diseño de lenguajes, específicamente de modelado, entre ellas “Alcance” (Scope) y “Nivel de Precisión” (Precision Level). Según Selic, estas características influyen en la capacidad del lenguaje para convertirse en un lenguaje ejecutable. El alcance define qué tan General o Especializado será el lenguaje. Por su parte, la generalidad viene a expensas de la expresividad y del detalle necesario para “Ejecutar Modelos” y “Generar Implementaciones Completas”. Mientras que la Especialización es una tendencia inevitable, lo cual se refleja en la ramificación constante de dominios de aplicación en subdominios cada vez más especializados y en que uno de los principios básicos de la Ingeniería Basada en Modelos (MDE) es elevar el nivel de abstracción de las especificaciones para acercarlas al dominio del problema. La Especialización entonces es una de las bases en la definición de un DSL. El Nivel de Precisión define qué tan Informal o Formal será el lenguaje. La característica de Lenguaje Ejecutable viene dada entonces de acuerdo a un nivel alto de formalidad, combinado con cierto nivel de precisión en su definición, consistencia interna y completitud. Selic define los Lenguajes Ejecutables como una subcategoría de lenguajes precisos que cubren un rango que incluye suficiente detalle para crear modelos ejecutables.

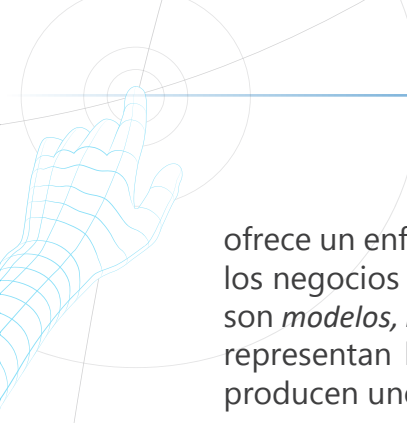
Con el fin de permitir la generación automática de modelos ejecutables, los modelos deben tener entonces unos ciertos niveles de especificación por medio de una sintaxis y una semántica definida con precisión. Un lenguaje muy difundido y generalizado para representar sintaxis de lenguajes es el MOF (*Meta Object Facility*, de OMG¹). Mediante el MOF es posible especificar familias de lenguajes que pueden integrarse entre sí. Uno de los lenguajes basados en MOF más difundidos es UML. Por tanto, MOF puede convertirse en el lenguaje base para la especificación de otras notaciones.

Por otro lado, MOF es una alternativa para desarrollar lenguaje (de modelado) de dominio específico. Aunque MOF es el estándar oficial, sólo existen algunas implementaciones, siendo la más popular Eclipse Modeling Framework (EMF). EMF se ha vuelto muy popular en el desarrollo de DSLs; un rango entero de plugins de Eclipse ha sido desarrollado para lidiar con los distintos aspectos del desarrollo de DSLs.

2.1.1.3.3 Model Driven Architecture (MDA)

MDA es un enfoque para el desarrollo, integración e interoperabilidad de sistemas de información. MDA propone una separación de modelos del sistema en tres capas, a saber, el modelo independiente de la computación (CIM), el modelo independiente de la plataforma (PIM), y el modelo específico de la plataforma (PSM). De acuerdo a esto,

¹ <http://www.omg.org/mof/>



ofrece un enfoque abierto y de tecnología neutral ante el reto del constante cambio de los negocios y de la tecnología (Miller y Mukerji, 2003). Los conceptos clave en MDA son *modelos*, *metamodelos* (que definen lenguajes abstractos por medio de los cuales se representan los lenguajes), y las *transformaciones* (que toman uno o más modelos y producen uno o más modelos a partir de estos) (ORMS, 2001).

MDA ha creado un gran interés, dado que asegura incrementar la productividad, la flexibilidad y la portabilidad de modelos de sistemas software (Miller y Mukerji, 2003). MDA define un enfoque que separa la especificación de la funcionalidad de un sistema de la especificación de la implementación de esa funcionalidad en una plataforma tecnológica específica. A su vez, define un enfoque para el desarrollo de software basado en modelado y *mapping* automático, a partir de los modelos hasta sus implementaciones (Bodart y Vanderdonckt, 1995).

La esencia de los fundamentos de MDA consiste en tres ideas, complementarias entre sí: (1) *Representación directa*: desplaza el foco del desarrollo de software lejos del dominio de la tecnología para dirigir las ideas y conceptos al dominio del problema; (2) *Automatización*: Propone usar herramientas computacionales para mecanizar el proceso de desarrollo del software y (3) *Estándares abiertos*: Los estándares han promovido el progreso de la tecnología, permitiendo la generación de un ecosistema de herramientas para propósitos generales y todo tipo de nichos especializados (Frankel, 2004a).

Aunque MDA soporta la construcción de modelos sofisticados en varios niveles de abstracción, éste no especifica reglas precisas o guías para explicar cómo puede ser usado por los ingenieros de software (Miller y Mukerji, 2003).

2.1.1.3.4 Enfoque de metamodelado

El metamodelado no solamente es la base para la especificación de los lenguajes de modelado tales como UML, sino también ofrece los fundamentos conceptuales para que la visión de la OMG pueda ser realidad (Küne, 2005). El metamodelado extiende y complementa las tecnologías existentes de modelado y orientación a objetos. Uno de sus conceptos fundamentales es la *clasificación*, la cual es aplicada mediante la relación "*instancia de*" entre los distintos niveles de metamodelado. Esta relación se usa para denotar la clasificación de los elementos en un nivel bajo, a partir de elementos de un nivel más alto. Un metamodelo es, por tanto, un modelo de clasificación. Es decir, un metamodelo es capaz de clasificar la información de los modelos que genera. Por tanto, la relación existente entre metamodelo y modelo es la de "*instancia de*". Un modelo, por otro lado, es un modelo de *tokens* a partir del cual no es posible clasificar otros modelos (Küne, 2005).

La OMG ha sugerido que una manera de hacer buen uso de UML en dominios particulares es por medio de los *UML profiles*. Esta técnica permite extender el metamodelo de UML. Sin embargo, la infraestructura de niveles de modelado definida para especificar UML está pensada como un framework para generar familias de



lenguajes (Atkinson et al, 2001). Esta ventaja del metamodelado es la clave para la definición de las familias de lenguajes asociadas a cada nivel o capa de un framework de modelos. Recientemente, se ha demostrado que el metamodelado puede ser usado para definir sintaxis concretas y sintaxis abstractas, así como semánticas (Kim et al, 2007).

La clasificación puede ser lingüística u ontológica, las cuales son clasificaciones ortogonales. La primera (lingüística) está relacionada con el hecho de que cada pieza de información necesita ser representada en un lenguaje a escoger y, por tanto, cada elemento usado puede ser clasificado de acuerdo a dicho lenguaje. Mientras que la clasificación ontológica tiene que ver con la relación que tiene la información con respecto a otros elementos dentro de un dominio específico de interés (Küne, 2005). El enfoque de metamodelo se está convirtiendo en una forma estándar de definir y manejar DSLs. Por tanto, puede ser usado para definir una interoperatividad entre los modelos diseñados y sus metamodelos.

2.1.1.3.5 Metodologías específicas del dominio


Un sistema complejo normalmente requiere de la utilización de varios lenguajes de modelado específico para describir su información. Aunque lenguajes como UML v.1.1 (Booch et al, 1998), IDEF (Cheng-Leong et al, 1999) y otros, han sido desarrollados como una colección de múltiples tipos de diagramas, la relación entre ellos aún sigue estando definida de forma pobre y poco clara. Esto da lugar a redundancias y pérdida de algunos subconjuntos de sus modelos. Por otro lado, otro inconveniente que plantea el uso de estos lenguajes de modelado de propósito genérico es que son demasiados generales para diseñar apropiadamente modelos en dominios específicos.

Las metodologías específicas del dominio (DSM) se enfocan en la especificación de lenguajes gráficos o textuales abstractos que se originan a partir del conocimiento e intereses de los expertos o desarrolladores de un dominio de negocio en particular. Esto les permite que puedan diseñar fácilmente sistemas desde sus propias perspectivas y puntos de vista, separando los conceptos de modelado del nivel de código fuente (Frankel, 2004b), (MetaCase, 2005).

En este sentido DSM se ajusta a los principios de la arquitectura dirigida por modelos (MDA) de la OMG, enfocándose en los modelos del sistema y generando transformaciones hasta llegar al código fuente (Kim et al, 2007). Sin embargo, mientras que MDA se centra en presentar la arquitectura y las guías para el metamodelado, DSM se centra en desarrollar libremente lenguajes gráficos abstractos, para que los desarrolladores puedan usarlos en sus dominios específicos. Una debilidad de DSM es que, en la actualidad, carece de mecanismos de integración (Kim et al, 2007).

2.1.1.3.6 Transformación de modelos

La transformación de modelos es el proceso que consiste en convertir un modelo en otro. Este proceso es de interés tanto para definir la sincronización entre modelos



pertencientes a una misma perspectiva y punto de vista, como entre ellas. Como la transformación de modelos es un área relativamente nueva, su taxonomía se define tardíamente y se dice que es un eslabón perdido de MDA (Frankel, 2004a). Algunos investigadores han tratado de clasificar y analizar varias técnicas de transformación en diferentes dominios (Kim et al, 2007). En la actualidad, se están llevando a cabo muchas investigaciones relativas a cómo expresar la transformación entre modelos. Estas transformaciones deben soportar varios puntos de vista de un sistema complejo. Las transformaciones automatizadas de modelos en MDA pueden proveer esta facilidad (Frankel, 2004a).

La transformación de modelos parece ser una de las operaciones más importantes a realizar sobre los modelos y una de las más críticas en MDE (Jouault y Kurtev, 2006). La transformación de modelos, pretende soportar la generación de un Modelo "b" (basado en un metamodelo "b") a partir de un Modelo "a" (basado en un metamodelo "a") (Atlas Group et al, 2005).

En estos entornos guiados por modelos se pretende que toda la información a manejar esté representada mediante modelos. Esto mismo es aplicable al caso de las transformaciones. Por tanto, una transformación debe ser representada mediante un modelo; el cual, a su vez, está basado en un metamodelo de transformación, que define la semántica de transformación. Bajo este paradigma, la OMG ha definido la especificación de un lenguaje de transformación denominado QVT (*Query/View/Transformation*) (ORMS, 2001). Esta especificación ha dado a lugar a ATL (*ATLAS Transformation Language*) (Jouault et al, 2006). ATL es un lenguaje de transformación, desarrollado como parte de una plataforma denominada AMMA (*ATLAS Model Management Architecture*), que incluye una herramienta sobre el entorno Eclipse.

2.1.1.4 Procesos de desarrollo de software

En la actualidad existen una gran cantidad de procesos de desarrollo de software que están consolidados en la comunidad de desarrollo de software. Dentro de los más representativos se encuentran el RUP (IBM_Rational, 2003a) y el SCRUM (Schwaber y Sutherland, 2010). El RUP es un proceso muy completo pero, a la vez, muy complejo y difícil de seguir en proyectos pequeños, por lo cual no será abordado en esta tesis. El SCRUM no es un proceso o una técnica para la construcción de productos; más bien, es un marco dentro del cual se puede emplear varios procesos y técnicas.

2.1.2 Casos de uso

El Caso de Uso (CU) es un elemento de modelado que ha sido ampliamente utilizado tanto en la ingeniería de software como en el diseño de interfaces de usuario (Constantine et al, 2001). Para comprender su estructura, es necesario dibujarlo en un diagrama. El Caso de Uso se especifica mediante una especificación de CU, donde se indica la interacción y la labor del sistema (Cockburn, 2000). Cada elemento de la



especificación de un CU, corresponde a un clasificador de información o token que describe la funcionalidad y cada aproximación metodológica sigue o propone una estructura diferente para el artefacto "*Especificación del Caso de Uso*". Ejemplos de clasificadores serían: "*Nombre del Caso de Uso*", "*Flujo Básico*", "*Precondición*", "*Postcondición*".

Los Casos de Uso se han catalogado como una herramienta de modelado que permite guiar el diseño y desarrollo de software, además de que se utilizan como punto de entrada a dichos procesos de desarrollo (Jacobson et al, 2011), (Nunes, 2001), (Lozano, 2001), (Giraldo, 2010), (Ng, 2002). Para esta investigación interesan los Casos de Uso porque se han definido como una excelente y poderosa forma de definir los requerimientos de comportamiento del software (Gottesdiener, 2002), porque permiten definir un proyecto y establecer un lenguaje común entre los stakeholders. De esta forma, los CU han tomado un interés tanto administrativo, como técnico en la industria de software.

En el aspecto administrativo, se observa por ejemplo, cómo Jacobson (Jacobson et al, 2011) presenta una evolución de la técnica de Casos de Uso enfocada en mejorar la gestión de proyectos de desarrollo de software, dejando un poco de lado los aspectos técnicos relacionados con la especificación de la funcionalidad de un producto software. Por otro lado, en el desarrollo centrado en el usuario se ha discutido mucho la eficacia y la utilidad de la técnica de especificación de Casos de Uso (Paternò, 2001), (Nunes, 2001), (Lozano, 2001), a pesar de que resulta ser una práctica muy útil para capturar requerimientos funcionales, dirigir la integración con la interfaz de usuario, el diseño de la funcionalidad y la colaboración (Giraldo, 2010).

La aproximación propuesta en Use-Case 2.0 (Jacobson et al, 2011) aborda el problema de la complejidad en el modelado con CU incorporando el concepto "*Use-Case Slice*" donde un "*Slice*" es una fracción de un CU que se diseña, implementa y prueba de forma separada, pero desde una perspectiva administrativa.

2.1.3 Workflow

El estudio de los Workflow es un área que incluye un conjunto de soluciones tecnológicas para la automatización o soporte de procesos de negocio o de trabajo descritos en un modelo de procesos. La tecnología Workflow proviene del campo de los llamados Sistemas de Información de Oficina.

La WfMC (Workflow Management Coalition) (WfMC, 1999) define Workflow como la automatización de un proceso de negocio, en su totalidad o en parte, en el que se pasan documentos de un participante a otro, información o tareas, de acuerdo a un conjunto de normas de procedimiento. Para hacer todo ese trabajo controlable y para fomentar la comunicación entre trabajadores, aparecen los sistemas de gestión de Workflow o WMS. Un WMS es un sistema que define, crea y gestiona la ejecución de uno o más Workflow a través del uso de software, que se ejecuta en uno o más motores de Workflow, y que es capaz de interpretar la definición del proceso,

interactuar con los participantes, en su caso, interactuar con herramientas informáticas y aplicaciones (WFMC, 1999).

Los Workflow pueden ser expresados en distintos lenguajes, por ejemplo: UML o BPM, siendo éste último el enfoque preferido en este campo. Sin embargo, una de las técnicas más difundidas, y en las que a menudo se basan el resto de propuestas encontradas en la literatura es el uso de Redes de Petri.

2.1.4 Teoría de la Actividad

La Teoría de la Actividad (TA) propuesta por Vygotsky (Vygotski, 1980), define la actividad humana como un sistema de diversos elementos y sus relaciones de mediación. Además, explora las relaciones sociales complejas inherentes a cualquier actividad humana. En su versión inicial, Vygotsky define tres elementos claves para estructurar la TA, "Sujeto", "Artefactos Mediadores" y "Objeto". A partir de estos elementos, Engeström define la extensión de la TA con seis elementos relacionados por medio de una estructura en forma de triángulos, conocida como Sistema de Actividad (Engeström, 1987). A partir de esta expansión, la TA se ha convertido en un marco teórico muy útil en el campo de la Interacción Humano Computador (HCI) (Nardi, 1996). La forma de un Sistema de Actividad (SA) desarrollada por Engeström se presenta en la Figura 2.1.

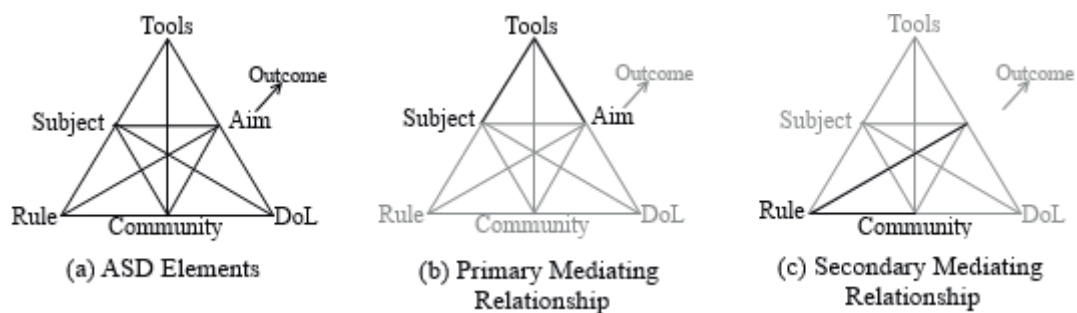


Figura 2.1. Estructura de un Sistema de Actividad (Engeström, 1987).

Engeström llama a esta estructura "Sistema de Actividad", a pesar de que dicha estructura sólo describe los elementos que conforman una sola actividad, mientras que un sistema normalmente describe o se compone de más que una sola actividad.

Elementos de un Sistema de Actividad (Engeström, 1987):

- **Aim/Object (Propósito/Objeto):** Cada actividad humana tiene un Propósito (físico o conceptual), por el cual la actividad es perseguida. Puede existir más de un propósito asociado con una sola actividad.
- **Outcome (Resultado):** Se produce como el resultado de la ejecución de la actividad.

- **Subject (Sujeto):** La actividad es realizada por el sujeto. Pueden existir múltiples sujetos ocupados en una sola actividad.
- **Tools (Herramientas):** El sujeto realiza la actividad usando una herramienta mediadora. Pueden existir múltiples herramientas utilizadas para realizar una sola actividad y las herramientas pueden ser físicas o conceptuales.
- **Community (Comunidad):** La comunidad es cualquiera que comparta el mismo propósito. Miembros de una comunidad pueden ser individuos, grupos de stakeholders, nombres de rol, etc.
- **Rules (Reglas):** Las relaciones entre el sujeto y la comunidad son mediadas por las reglas. Pueden ser implícitas o explícitas y representan normas y convenciones presentes en la actividad.
- **Division of Labor (División de la Labor):** Las relaciones entre el propósito y la comunidad son mediadas por la División de la Labor (DoL), que especifica cómo se distribuye la tarea de alcanzar el propósito a través de la comunidad.


Mediación:

Los elementos estructurales de un Sistema de Actividad contienen *Elementos Mediadores (tool, rule y DoL)* y *Elementos Mediados (subject, aim y community)*. Una relación de mediación (mediating relationship) se representa como las líneas entre dos elementos mediados y que pasan a través de un elemento mediador. Una relación de mediación causa algún resultado, por ejemplo, la habilidad del sujeto para alcanzar el propósito de la actividad utilizando una herramienta (Figura 2.1(b)) o la habilidad de la comunidad para alcanzar el propósito de la actividad cumpliendo un conjunto de reglas (Figura 2.1(c)). Se definen también relaciones de mediación primaria y secundaria. Las relaciones de mediación primarias son: (1) una herramienta media la interacción entre un sujeto y el propósito de una actividad, (2) las reglas median interacciones entre comunidad y sujeto, y (3) la división de la labor media interacciones entre la comunidad y el propósito de la actividad. Posibles relaciones de mediación secundarias serían las que se describen como líneas adicionales en el Sistema de Actividad, por ejemplo: (1) una regla puede mediar entre la comunidad y el propósito de la actividad (Figura 2.1(c)) o entre un sujeto y el propósito de la actividad. Según (Georg et al, 2015), las relaciones de mediación primarias son suficientes para describir una actividad.

Creación de redes

La TA reconoce que la actividad humana pocas veces se lleva a cabo de manera aislada. Soporta entonces el concepto de redes de sistema de actividad donde se conectan múltiples SA en el sentido de que el resultado de un SA puede usarse como uno de los demás tipos de elementos en otros SAs. El uso más común de esta relación es cuando el resultado de una actividad, se utiliza como una herramienta de otra. Otro ejemplo sería cuando un resultado puede ser una regla aplicable a otro SA o el





entrenamiento que suministraría un sujeto para otra actividad.

Evolución:

La TA también propone que las contradicciones existen en varios niveles dentro de la actividad humana y que son estas contradicciones las que permiten la evolución de esa actividad. La evolución ocurre para disminuir contradicciones dentro y entre elementos y previas versiones de una actividad. En particular, Engeström identifica cuatro tipos de contradicciones: (1) dentro de un elemento de la TA, por ejemplo, entre una regla y otra en una actividad, (2) entre elementos en una actividad, por ejemplo, entre un ítem de DoL y una regla de la misma actividad, (3) entre múltiples actividades en red, por ejemplo entre el resultado de una actividad y otra actividad que usa ese resultado como una herramienta, y (4) entre múltiples evoluciones, por ejemplo, entre una actividad ad-hoc y una actividad más regulada dirigida a la repetibilidad de esa actividad.

Principalmente, para el desarrollo de esta investigación, interesa la Teoría de la Actividad por sus diversas aplicaciones en la Interacción Humano Computador y en la Ingeniería de Software (Collins et al, 2002), (Constantine, 2009), (Hasan, 1998), (Kaptelinin et al, 1999), (Korpela et al, 2000), (Neto et al, 2005a). Como se comenta en (Georg et al, 2015), la TA puede ser útil en situaciones donde es necesario explorar el contexto social diverso y complejo de un sistema, particularmente aquellas que tienen componentes humanos y de cómputo.

Adicionalmente, la TA y HCI tienen puntos de intersección que se reflejan en la participación de actores en actividades y en la naturaleza jerárquica del rendimiento de las actividades. La TA provee una forma coherente de entender y modelar actores como usuarios de herramientas acopladas con otros participantes y artefactos (Constantine, 2009).

2.1.5 Ingeniería semiótica (Souza, 2005)

La Ingeniería Semiótica es una teoría de HCI, donde la comunicación entre diseñadores y usuarios en tiempo de interacción, es el enfoque de investigación. Los diseñadores de alguna manera tienen que estar allí para indicar a los usuarios cómo interactuar con las señales que han inventado para determinada aplicación software. Esta teoría hace muchas conexiones entre los conceptos tomados de la semiótica y las ciencias de la computación y considera de manera diferente al HCI si se compara con las teorías existentes de tendencia cognitiva. La Ingeniería Semiótica está pensada para ayudar a los diseñadores a comunicar su visión a los usuarios, de formas mucho más fáciles. Y tal vez entonces los usuarios empezarán a entender los diseños, y no sólo luchar para aprender cuáles controles activar.

Como su nombre lo indica, se refiere a la semiótica y a la ingeniería para construir una explicación teórica completa desde el HCI. La semiótica es importante porque el HCI involucra significación y procesos de significado-relacionado que toma lugar tanto en

los sistemas de cómputo como en las mentes humanas. Y la ingeniería es importante porque se espera que la teoría soporte el diseño y la construcción de artefactos. Por otro lado, la semiótica y la ingeniería se acoplan estrechamente cuando pensamos que los artefactos de HCI son construcciones intelectuales, es decir, el resultado de elecciones y decisiones guiadas por el razonamiento, creación de sentido y de competencias técnicas, en lugar de leyes naturales previsibles.

La Figura 2.2 muestra una comparación esquemática entre el Diseño Centrado en el Usuario (DCU) y la Ingeniería Semiótica. En el lado del DCU, la imagen del sistema es el único rastro de todo el trabajo intelectual que ha tomado lugar durante el diseño, y es lo que se requiere que el usuario aprenda. Mientras que en el lado de la Ingeniería Semiótica, el diseñador mismo está presente en tiempo de interacción, contándole al usuario acerca de su visión de diseño, a la cual el usuario responderá de varias formas (incluyendo formas inesperadas y creativas).

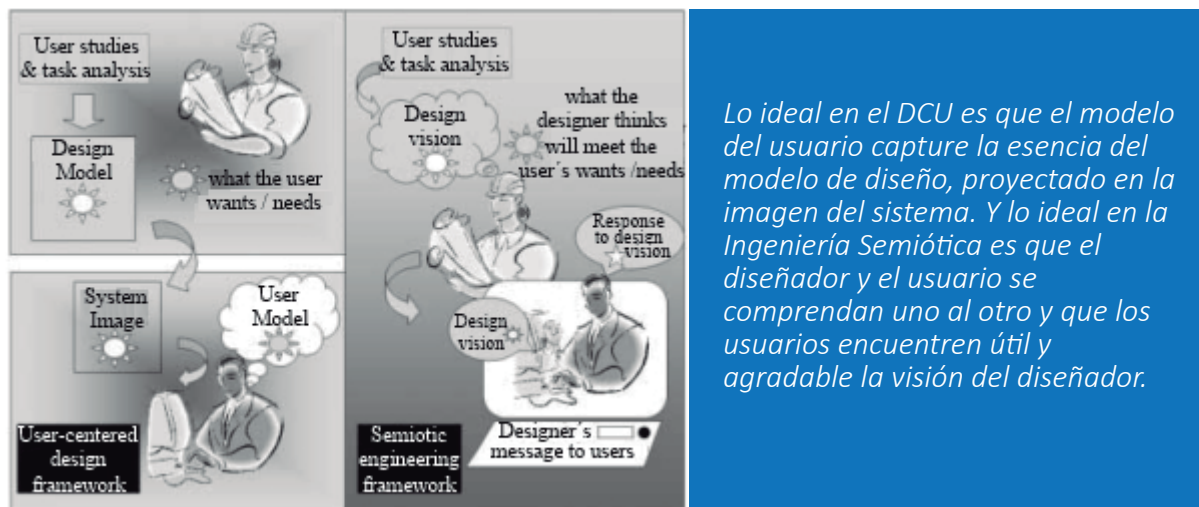


Figura 2.2. Comparación Esquemática entre el DCU y la Ingeniería Semiótica (Souza, 2005)

2.1.6 Análisis de la comunicación

El Análisis de la Comunicación es un método propuesto por España (España et al, 2009), para el desarrollo y computarización de los Sistemas de Información Empresariales. Se enfoca en las interacciones comunicativas que ocurren entre el Sistema de Información y su ambiente, a partir de la especificación de los requisitos del sistema.

En el Análisis de la Comunicación, las especificaciones de requisitos están organizadas principalmente alrededor del conjunto de interacciones comunicativas que los actores de la empresa necesitan realizar para llevar a cabo sus tareas (González et al, 2011). La propuesta metodológica presenta una estructura de requisitos por capas que cubre los espacios del problema y de la solución (Figura 2.3).

El análisis de la comunicación ofrece técnicas para análisis de la interacción, actividades 2, 4 y 6, y para el análisis de objetos de negocio, actividades 3, 5 y 7, lo cual facilita el

tratar con la dualidad entre los aspectos estáticos y dinámicos en los sistemas organizacionales. En los niveles L1 y L2, el análisis busca refinar un sistema complejo dentro de sub-sistemas y obtener el repertorio de interacciones comunicativas que los usuarios necesitan para su práctica en el trabajo. Desde el nivel L2 al nivel L3 el analista hace un salto "cuántico" e inicia especificando las interacciones comunicativas identificadas. Desde el nivel L3 hacia el nivel L4, el analista cambia el enfoque de la especificación desde una perspectiva pragmática a una perspectiva semántica y sintáctica; el modelo de requisitos añade detalles acerca de cómo los mensajes se muestran y se editan, por ejemplo, interfaces de diseño. El nivel L3 se enfoca en los mensajes o datos en movimiento, mientras que el nivel L4 se enfoca en la memoria del sistema de información o datos en reposo, con el fin de asegurar la persistencia de las interacciones comunicativas. Finalmente, se diseña la arquitectura tecnológica y se implementa la aplicación software.

Se observa en la Figura 2.3 una estructura de dos dimensiones. Una dimensión relacionada con la dualidad estática-dinámica (eje horizontal). Y otra dimensión relacionada al refinamiento (eje vertical).

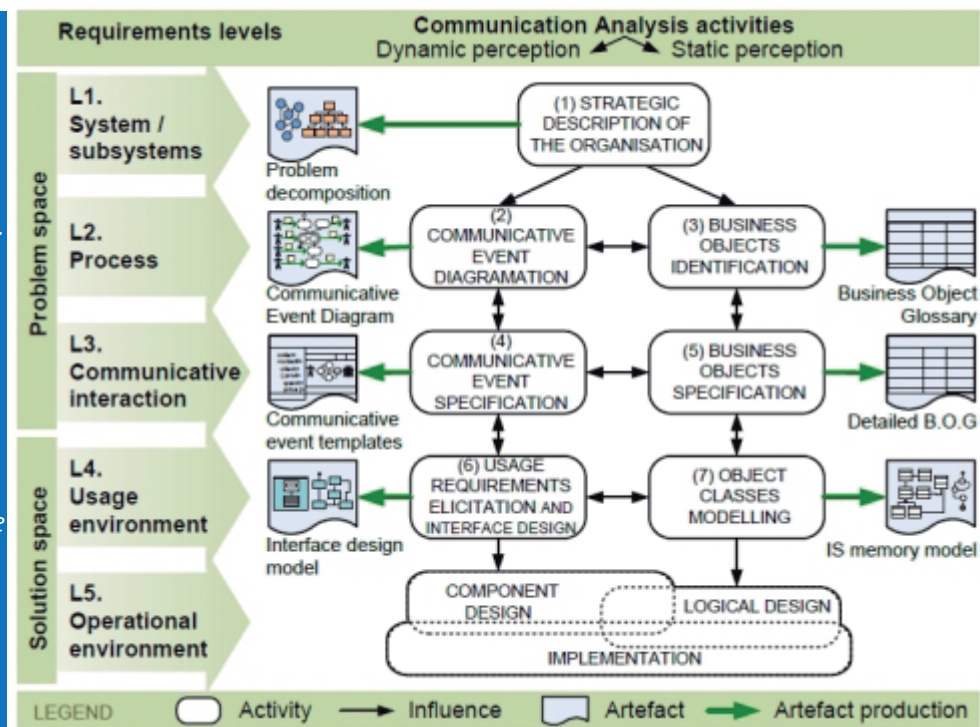


Figura 2.3. Workflow para la estructura de requisitos y Análisis de la Comunicación (González et al, 2011)

2.1.7 Interacción humano computador (HCI)

La Interfaz de Usuario (IU) es la parte del sistema que se ve, oye y siente. Otras partes del sistema están ocultas (por ejemplo, las base de datos en las que se almacena la información). Aunque los usuarios no ven las partes ocultas, se imaginan, en cierta medida lo que está pasando "detrás de la pantalla" (Lauesen, 2005). Esta comprensión de lo que sucede, con frecuencia es bastante importante y es la preocupación más relevante en el desarrollo de los sistemas interactivos. Cuando se utiliza un




computador normalmente se hace por medio de órdenes o comandos que, por lo general, se llevan a cabo utilizando los dispositivos estándar de entrada-salida, por ejemplo: teclado, mouse, pantalla, etc. En este proceso, el computador responde a las órdenes mediante estos dispositivos. A veces es al contrario, el computador emite comandos y el usuario interviene para completar una orden. Este proceso se conoce como *interacción humano computador* (HCI), y se lleva a cabo sobre lo que se conoce como *sistema interactivo*.

Según la SIGCHI (SIGCHI, 1992) (*Special Interest Group in Computer Human Interaction*), la interacción humano computador (HCI) es la disciplina relacionada con el diseño, evaluación e implementación de sistemas informáticos interactivos para el uso de seres humanos, y con el estudio de los fenómenos más importantes con los que está relacionado. HCI es un campo multidisciplinar que se interesa por todos los aspectos relacionados con el proceso de interacción entre los usuarios y los sistemas interactivos. Para esto, necesita estudiar y conocer los individuos como parte integrante de grupos u organizaciones desde varias áreas de conocimiento, tales como, la Psicología, la Sociología, la Ingeniería del Software, la Ergonomía, la Inteligencia Artificial, etc.

La interfaz de usuario es el punto de conexión entre los seres humanos y los dispositivos. En un PC estándar, la interfaz de usuario se compone de la pantalla, teclado, ratón y altavoces. En sistemas más avanzados, la interfaz puede incluir la entrada de voz a través de un micrófono, botones especiales, luces y pantallas. También se puede hablar de la interfaz de usuario de una fotocopiadora, un televisor o un automóvil. La interfaz de usuario de un sistema consiste en aquellos aspectos del sistema con los que el usuario entra en contacto, físicamente, perceptivamente y conceptualmente. Los aspectos del sistema que están escondidos para el usuario se denominan implementación (Moran, 1981). Desde el punto de vista del usuario no interesa la forma como se utilizan las funcionalidades y, por lo general, el usuario no es consciente de aquellas tareas que la interfaz le oculta. Tan sólo interacciona con el computador para poder realizar una serie de tareas y solamente está interesado en los resultados de las mismas, mostrados precisamente por dicha interfaz (Granollers, 2004).

La interacción con el usuario debe llevarse a cabo por medio de interfaces que aumenten la productividad y disminuyan el tiempo de aprendizaje del sistema. De igual forma, deben considerar la diversidad en múltiples niveles, diferentes tipos de usuarios, aspectos culturales, físicos, cognitivos, etc. Intentar incluir estos requerimientos es una de las preocupaciones de HCI. La interfaz de usuario debe ser entonces útil, efectiva, eficiente y usable (Diaper, 1989). El enfoque en el desarrollo de estos factores y las técnicas para medirlos se conoce como *usabilidad* (Nielsen, 1993). Se puede considerar la usabilidad como la característica que hace que un software sea más fácil de utilizar y aprender por parte del usuario. Las aplicaciones usables son aquellas que permiten al usuario centrarse en la tarea que están realizando y no en la aplicación utilizada para llevarla a cabo. Por lo tanto, los factores principales que deben considerarse al hablar de usabilidad son la facilidad de aprendizaje, la efectividad de



uso y la satisfacción con las que las personas son capaces de realizar sus tareas gracias al uso del producto con el que se está trabajando (Shneiderman, 2004). Establecer unos principios de diseño en ingeniería basados en la usabilidad puede redundar en una reducción de los costes de producción, de mantenimiento y de uso, así como en una mejora de la calidad del producto.

2.2 ANTECEDENTES

El Modelado de la actividad puede ser visto como una manera de formalizar los resultados de un análisis de tareas, cuyo objetivo es entender y captar la naturaleza y estructura de la organización, su entorno, los objetivos y sus actividades (Giraldo, 2010). Para crear sistemas interactivos eficaces es necesario entender la naturaleza y la estructura del trabajo, es decir, cuál es el trabajo que se debe hacer y cómo se debe llevar a cabo (de acuerdo a los objetivos de la organización y sus metas). A continuación se describen algunas propuestas que fueron analizadas para determinar la forma en la que modelan y ejecutan la actividad en el desarrollo de sistemas interactivos, de acuerdo con las **estrategias 1 y 2**, planteadas en la sección 1.2.2. El análisis está enfocado hacia los artefactos que guían el proceso de desarrollo, la notación y las herramientas que se utilizan hasta generar el producto final, sistema funcional o interfaz de usuario. Las propuestas analizadas se clasifican desde las perspectivas de la funcionalidad, colaboración, y la interfaz de usuario.

2.2.1 Modelado y ejecución de la actividad en sistemas interactivos, funcionalidad

En esta sección se presentarán los principales estudios encontrados en la literatura relacionados con el modelado de la actividad en sistemas interactivos, y que se enfocan en la **funcionalidad**. Luego de su descripción, se presenta un breve análisis para cada uno de ellos. El análisis de cada propuesta sirve como referencia para orientar la investigación y resulta útil para identificar las ventajas y desventajas de las propuestas existentes.

2.2.1.1 OpenUp

OpenUP es una personalización del RUP que adopta sus principales características. Es un proceso de desarrollo de software que es mínimo, suficiente, completo y extensible en el sentido de que éste puede manifestarse como un proceso completo para construir un sistema. Está guiado por casos de uso y escenarios y posee un enfoque centrado en la arquitectura para definir el desarrollo (Balduino, 2007). Los casos de uso proporcionan orientación para iniciar el diseño, tanto en la arquitectura de negocio como en la arquitectura de software. Además, OpenUP es extensible, lo cual permite utilizar su repositorio de métodos para adaptarlo y añadir contenido, según sea necesario.

OpenUp está organizado en seis disciplinas. Una *disciplina* es un conjunto de tareas que están relacionadas con un área grande de enfoque dentro del proyecto global. Dentro



del ciclo de vida, las tareas se realizan simultáneamente en varias disciplinas. La separación de las tareas en distintas disciplinas es una manera eficaz de organizar el contenido para hacer más fácil la comprensión del proceso.

2.2.1.1.1 Análisis de la propuesta OpenUp

El desarrollo de productos software mediante el proceso OpenUp está dirigido por Casos de Uso. Este proceso se ha enfocado en el desarrollo de la funcionalidad, siguiendo la línea de interés de la Ingeniería de Software. Es de interés el OpenUp para el desarrollo de esta tesis, ya que es lo suficientemente pequeño y compacto y principalmente porque especifica la actividad mediante el lenguaje UML, una de las notaciones más utilizadas por la comunidad de desarrolladores de software para el modelado de sistemas interactivos. A pesar de esto, dicha notación no permite capturar de manera adecuada y expresiva, distintas representaciones de la labor en los sistemas interactivos (Rourke, 2002), (Paternò, 2001), (Cesare et al, 2006). Ejemplos de esto se encuentran: en el modelado de las tareas, para diferenciar la labor del usuario y del sistema, en el modelado de la comunicación y la coordinación, el modelado de los objetivos de una manera estructurada, etc.

2.2.1.2 SCRUM

El SCRUM no es un proceso o una técnica para la construcción de productos; más bien, es un marco dentro del cual se puede emplear varios procesos y técnicas. El papel de Scrum es aflorar la eficacia relativa de sus prácticas de desarrollo de manera que se puede mejorar sobre ellos, al tiempo que proporciona un marco dentro del cual se pueden desarrollar productos complejos. SCRUM se basa en la teoría de control de proceso empírico, emplea un enfoque iterativo e incremental para optimizar la previsibilidad y control de riesgos. Cada implementación del control del proceso empírico se basa en tres pilares que son Transparencia, Inspección y Adaptación.

2.2.1.2.1 Análisis de la propuesta SCRUM

Al igual que el OpenUp, el SCRUM está fuertemente orientado al desarrollo de la Funcionalidad. Está dirigido por iteraciones llamadas Sprints donde se producen fragmentos funcionales, releases, del producto final. La especificación de la actividad se hace mediante historias de usuario, que a diferencia de los Casos de Uso, no tienen notación gráfica o lenguaje de modelado asociado; aunque sí tiene una estructura sintáctica bien definida mediante frases escritas en lenguaje natural. Es posible que esta estructura sintáctica sirva de base para la definición de clasificadores en la Taxonomía de la Actividad.

2.2.1.3 BPMN

Business Process Model and Notation (BPMN) es una de las técnicas más utilizadas para el modelado de procesos de negocio. Lo que pretende BPMN es crear un puente estandarizado para la brecha que existe entre el diseño de procesos de negocio y la



implementación de procesos (OMG, 2011).

La descripción del metamodelo de BPMN incluye la representación de los modelos de procesos de negocio traducidos al lenguaje BPEL, que es un lenguaje basado en XML para describir procesos de negocio en el cual la mayoría de tareas representan interacciones entre los procesos y servicios Web externos (OMG, 2012). El proceso BPEL por sí mismo se representa como un servicio Web y es realizado por un motor BPEL que ejecuta la descripción del proceso.

2.2.1.3.1 Análisis de la propuesta BPMN

BPMN se describe como una técnica de modelado, así que no se evidencia que tenga asociado algún proceso de desarrollo de software específico, como sí lo tiene por ejemplo, el lenguaje UML, con el proceso unificado de Rational (RUP). BPMN se enfoca bastante en los elementos de modelado que describen los controles en el flujo de los procesos, y en este estudio se clasifica como una notación enfocada en el desarrollo de la funcionalidad de sistemas interactivos. Puede ser útil en la definición de clasificadores para la Taxonomía de la Actividad, en el nivel de abstracción de Negocio.

2.2.1.4 Diagramas de estados

El lenguaje de modelado UML utiliza el concepto de "Estado" para definir el comportamiento de un objeto. Un estado es una condición ontológica que persiste por un periodo de tiempo significativo. El UML representa estados de dos maneras: diagramas de estados (statecharts) y diagramas de actividad. Las máquinas de estado de Harel (Harel, 1987) forman la base de los diagramas de estados. Estos diagramas son un formalismo visual que describe un espacio de estado de un objeto y son particularmente apropiados para representar sistemas reactivos, es decir, sistemas que esperan por y reaccionan a eventos asíncronos.

Un diagrama de actividad es un tipo de máquina de estado en el cual la mayoría o todas las transiciones entre estados ocurren no por los eventos asíncronos, sino siguiendo la completitud de las actividades realizadas mientras se ejecuta la actividad misma. Por esta razón, los diagramas de actividad resultan más apropiados para mostrar principalmente comportamiento síncrono.

2.2.1.4.1 Análisis de la propuesta diagrama de estados

En este trabajo ubicamos los diagramas de estados en el nivel de abstracción de Sistema y se toman como un elemento clave en la generación de código ejecutable de los objetos que conforman el sistema interactivo. Los clasificadores que son útiles para la taxonomía están relacionados con las categorías del "Cómo" (Estados, y Acciones), del "Por qué" (Guardas) y del "Tiempo" (Eventos).

Los diagramas de actividad los catalogamos en el nivel de abstracción de Negocio, debido a su utilidad para capturar información en los modelos de actividad de negocio complejos.



2.2.1.5 Redes de Petri

Las redes de Petri son herramientas de modelado gráfico y modelado matemático, aplicables a muchos sistemas. El concepto de red de Petri tiene sus orígenes en a disertación de la tesis doctoral de Carl Adam Petri, en 1962. Como herramienta gráfica, las redes de Petri se pueden utilizar como ayuda en la comunicación visual, similarmente a los diagramas de flujo, diagramas de bloques, y redes. Se utilizan los Tokens a través de las redes para simular las actividades dinámicas y concurrentes de los sistemas. Como herramienta matemática, es posible configurar ecuaciones de estado, ecuaciones algebraicas, y otros modelos matemáticos que controlan el comportamiento de los sistemas (Garcia, 2010).

2.2.1.5.1 Análisis de la propuesta redes de petri

Las Redes de Petri resultan interesantes para este trabajo porque proporcionan una definición formal, a diferencia de los diagramas de estados en UML, que evita ambigüedades, incertidumbres y contradicciones, además de que pueden ser visualizadas gráficamente.

2.2.1.6 OO Method

OO-Method es un método orientado a objetos para describir un esquema conceptual de sistemas de información por medio de diagramas compatibles con UML, y que se basa en el lenguaje formal OASIS.

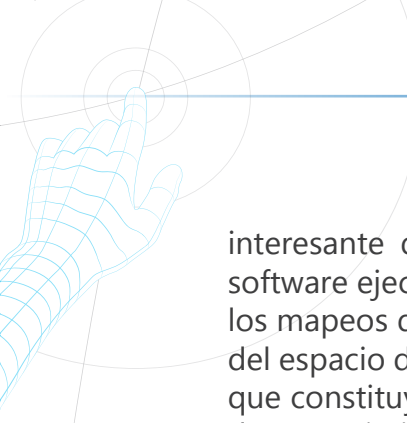
Pastor, (Pastor et al, 2001) describe el OO-Method como un ambiente para la Ingeniería de Requisitos Asistida por Computador (CARE), donde se enfoca en cómo capturar apropiadamente los requerimientos del sistema con el fin de administrar el proceso completo de producción de software.

El modelo conceptual resultante especifica lo que el sistema es (espacio del problema). Luego se provee un modelo de ejecución abstracta para guiar la representación de esos requerimientos en un ambiente de desarrollo de software específico (espacio de la solución, la cual se centra en cómo será implementado el sistema).

El modelo de ejecución abstracto se basa en el concepto de Construcciones de Modelado Conceptual (Conceptual Modeling Constructs). El OO-Method provee una representación software bien definida de dichas construcciones en el espacio de la solución. Se introduce también un modelo de ejecución concreto basado en una arquitectura basada en componentes, para hacer frente a las peculiaridades de los sistemas basados en componentes.

2.2.1.6.1 Análisis de la propuesta OO Method

La propuesta OO-Method se ha catalogado en este análisis como un método que se enfoca en el desarrollo de la Funcionalidad para productos software. Resulta



interesante que la propuesta está orientada a la generación de componentes de software ejecutables de una forma automatizada, por medio de la implementación de los mapeos desde los conceptos del espacio del problema, hacia las representaciones del espacio de la solución. De tal manera que se obtiene un conjunto de componentes que constituyen un producto software funcionalmente equivalente a la especificación de requerimientos elicitados en el paso del modelado conceptual. Esto puede aportar significativamente en la definición del componente relacionado con los lenguajes ejecutables que se propone en este trabajo.

Adicionalmente, el Modelo de Ejecución del OO-Method está soportado por una herramienta de desarrollo basada en componentes e implementada sobre Java, llamada JEM. El propósito de esta herramienta es animar los modelos conceptuales capturados con el OO-Method, sobre ambientes distribuidos internet/intranet.

2.2.1.7 Whitewater Interactive System Development with object models (WISDOM)

Wisdom (Nunes, 2001), es una aproximación metodológica basada en UML que proporciona una conexión entre el desarrollo de la interfaz de usuario y la funcionalidad. Wisdom ofrece soporte al desarrollo centrado en el usuario y al diseño de la interfaz de usuario. Proporciona mecanismos para abordar el modelado de los roles de usuario, modelado de la interacción, diálogo y presentación. Este método integra el análisis de requisitos basado en casos de uso con el modelado esencial para llevar a cabo una diferenciación de las actividades relacionadas con el usuario y con el sistema.

2.2.1.7.1 Análisis de la propuesta WISDOM

En Wisdom se explora cómo el modelado de objetos puede contribuir a integrar la ingeniería de la usabilidad en el desarrollo de software proporcionando beneficios para ambas disciplinas. Para esto, introduce nuevos elementos para especificar la interfaz de usuario haciendo uso de un subconjunto de UML. A su vez, propone un modelo arquitectónico para reducir el número total de los conceptos necesarios para desarrollar sistemas interactivos.

Se observa que en el modelo de interacción relaciona los elementos "vista" y "tarea" y en el modelo de diálogo utiliza el mismo clasificador o elemento de modelado para "tarea". Esto puede ser útil como una característica a mejorar en la selección de clasificadores de la actividad.

Se ha catalogado esta propuesta como enfocada a desarrollar la funcionalidad y la interfaz de usuario.

2.2.1.8 Model Driven Software Development Framework

El Framework para Desarrollo de Software Dirigido por Modelos, se basa en el Análisis de la Comunicación (Cubillo, 2011). Es un método que tiene como objetivo facilitar la interpretación y representación de un dominio como un sistema de información y

(posiblemente) el desarrollo de un sub-sistema de información computarizado. Analiza las interacciones comunicativas entre el sistema de información y su ambiente.

El núcleo metodológico del análisis de la comunicación es la etapa de análisis del sistema de información, cuyo resultado es una especificación del análisis, una documentación orientada a la comunicación que describe el sistema de información sin tener en cuenta su posible computarización (ver Figura 2.4). De esta manera, la especificación de análisis producida por el Análisis de la Comunicación corresponde a la capa CIM (Computation Independent Model) de la Arquitectura Dirigida por Modelos. Esta especificación incluye una descomposición del sistema que tiene la intención de reducir la complejidad, un conjunto de diagramas de eventos comunicativos que describen procesos de negocio desde una perspectiva comunicacional, un conjunto de plantillas de especificaciones de eventos que describen cada actividad de negocio en detalle, y los requisitos de objetos que especifican los objetos de negocio desde la perspectiva del usuario.

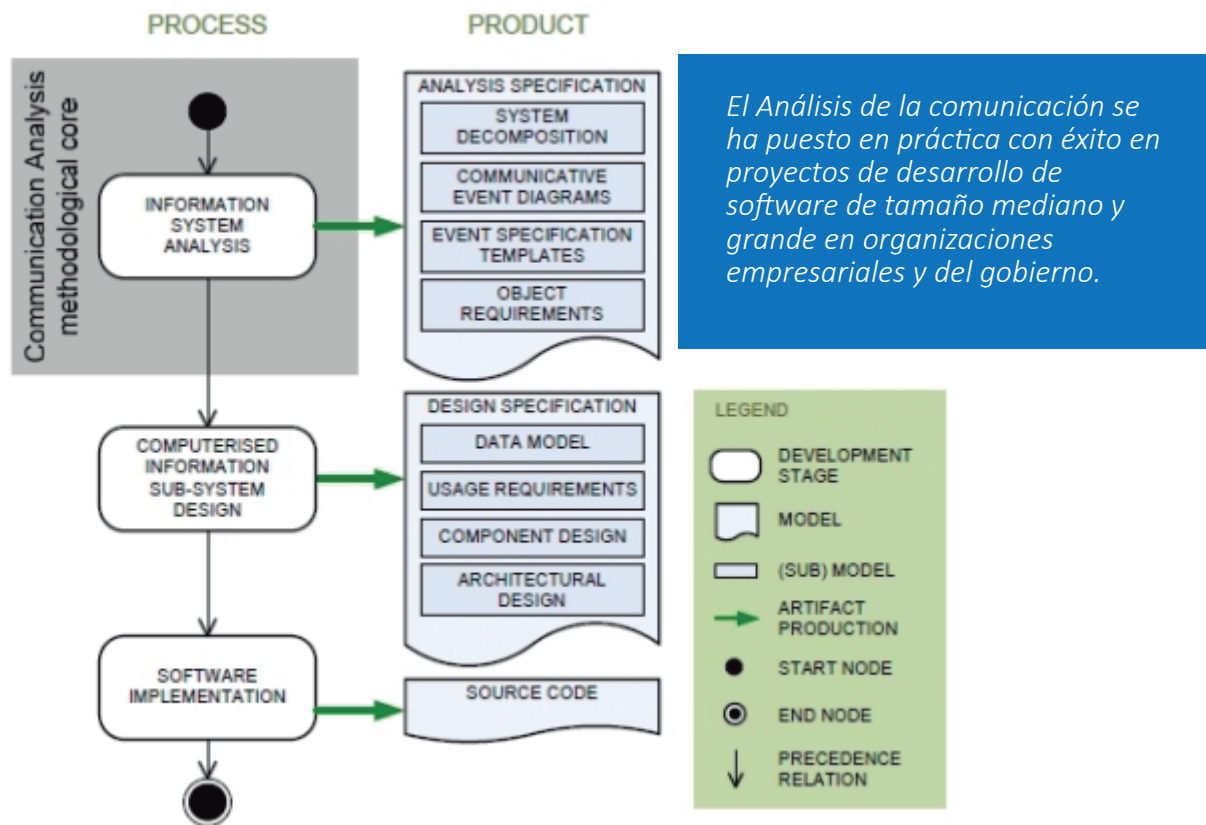


Figura 2.4. Base metodológica del Análisis de la Comunicación con un amplio método de desarrollo (España, 2011)

Existen pocas herramientas de prototipado, pero los practicantes en Ingeniería de Requisitos utilizan principalmente herramientas de diagramación de propósito general, por ejemplo, Visio © y procesadores de texto, por ejemplo, Microsoft Word © con el fin de especificar los requisitos. Las etapas posteriores, como el diseño,

implementación y pruebas, aunque impulsadas por la especificación de requisitos, no hacen uso de las tecnologías de transformación de modelos.

2.2.1.8.1 Análisis de la propuesta model driven software development framework

Esta propuesta resulta muy interesante porque define un marco conceptual muy completo sobre el modelado de la comunicación que tiene lugar en la especificación de los sistemas de información, a nivel de negocio (nivel independiente de la tecnología). Para el presente estudio se considera que la expresividad de la notación utilizada es suceptible de mejoras y que puede resultar muy útil en la definición de clasificadores y de elementos del metamodelo para la TxA. Adicionalmente, el hecho de que la propuesta aplique los principios de MDE, hace que se convierta en una de las propuestas más relevantes para este trabajo. Se ha catalogado esta propuesta como enfocada en el desarrollo de la funcionalidad para sistemas interactivos.

2.2.2 Modelado y ejecución de la actividad en sistemas interactivos, colaboración

En esta sección se presentarán los principales estudios encontrados en la literatura relacionados con el modelado de la actividad en sistemas interactivos, y que se enfocan en la **colaboración**. Luego de su descripción, se presenta un breve análisis para cada uno de ellos. El análisis de cada propuesta sirve como referencia para orientar la investigación y resulta útil para identificar las ventajas y desventajas de las propuestas existentes.

2.2.2.1 Collaborative Interactive Applications Methodology (CIAM)

CIAM (Molina et al, 2006) es una aproximación metodológica para el diseño de la interfaz de usuario en aplicaciones colaborativas. Las etapas que componen la propuesta CIAM se muestran en la Figura 2.5. Se observa que las primeras etapas abordan un modelado centrado en el grupo, pasando en fases posteriores a un modelo más centrado en el proceso (cooperativo, colaborativo y de coordinación), acercándose, a medida que se baja en el nivel de abstracción, hacia un modelado centrado en el usuario, en el que se modelan las tareas interactivas (es decir, el diálogo que se da entre un usuario individual y la aplicación).

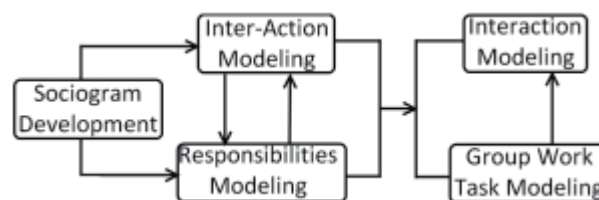


Figura 2.5. Etapas de la propuesta metodológica CIAM (Molina et al, 2006) .

El primer paso es la **Creación del Sociograma** que representa la estructura de la organización, así como las relaciones que existen entre los distintos integrantes de la misma. El segundo paso es el **Modelado de la Inter-Acción**, que consiste en la especificación de las principales tareas (o procesos orientados a objetivos específicos) que definen el trabajo en grupo que se desarrolla en el seno de la organización. Luego de la especificación del trabajo en grupo, se extiende el conjunto de responsabilidades a cada rol, adicionando asignaciones individuales (**Modelado de la responsabilidad**). La información detallada en esta fase se complementa con el modelado de la inter-acción, siendo necesario que ambos modelos sean coherentes entre sí. El siguiente paso consiste en extender los detalles de las tareas cooperativas y colaborativas previamente indentificadas (**Modelado del Trabajo en Grupo**). Para el modelado de las tareas cooperativas se emplea el *grafo de descomposición de responsabilidades*, en el que se detallan las subtareas que componen una tarea cooperativa, de forma que, en el nivel más bajo de abstracción se contará exclusivamente con tareas individuales. Para el modelado de las tareas colaborativas se incluye la especificación de los roles participantes en su ejecución y de los objetos del modelo de datos manejados por el equipo de trabajo. La notación empleada incluye la especificación del *contexto compartido*. Finalmente, en el **Modelado de la Interacción** se modelan los aspectos puramente interactivos de la aplicación, refiriéndonos exclusivamente a aspectos de interacción persona-computador. Se crea un modelo de interacción para cada tarea de naturaleza individual identificada en las distintas fases del proceso de refinamiento gradual que propone CIAM. Para cada una de ellas se crea un árbol de descomposición de tareas interactivas en notación CTT. En cuanto a las tareas colaborativas, el modelo de interacción se obtiene directamente a partir de la definición del contexto compartido.

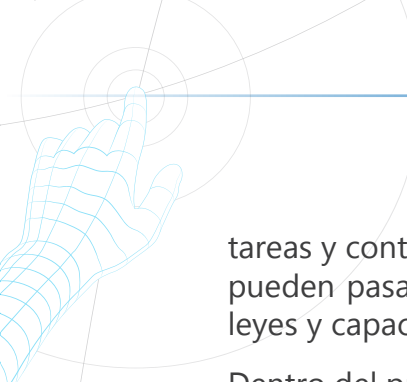
2.2.2.1.1 Análisis de la propuesta CIAM

La propuesta metodológica de CIAM describe sus artefactos mediante la notación CIAM (*Collaborative Interactive Application Notiation*). En particular, es de interés para este trabajo que CIAM haga distinción entre el Modelo de Inter-Acción y el Modelo de Interacción. Siendo el primero el que especifica las principales tareas, que definen el trabajo en grupo, que se desarrollan en el seno de la organización. Y el segundo, el que especifica las tareas que tienen lugar entre el usuario y el sistema interactivo. Se basa en la notación CTT y está asociado a las tareas individuales. La propuesta CIAM se ha clasificado dentro de las propuestas que se interesan por desarrollar la Interfaz de Usuario en los sistemas interactivos, con el objetivo de mejorar el aspecto de la Colaboración.

2.2.2.2 A methodology for analysis and design of cooperative systems (AMENITIES)

AMENITIES (Garrido, 2003) es una propuesta metodológica de especificación y diseño de sistemas cooperativos basada en modelos de comportamiento y tareas. Dicha propuesta incluye la definición de un marco conceptual y metodológico que consta de un conjunto de modelos y fases a seguir. Esta metodología está basada en modelos de





tareas y contempla aspectos dinámicos, tales como situaciones en las que los usuarios pueden pasar de desempeñar un rol a desempeñar otro, por medio de una serie de leyes y capacidades.

Dentro del proceso de desarrollo se identifican tres fases principales. Una primera fase de obtención y representación del modelo de requisitos, donde se hace uso de técnicas como la etnografía aplicada o los casos de uso. En una segunda fase, se crea el llamado modelo cooperativo, para lo que se hace uso de la notación COMO-UML. La última fase permite conectar el modelo cooperativo con el diseño inicial del sistema, así como el paso a métodos formales (Redes de Petri Coloreadas), que permiten el análisis y la verificación de los modelos, así como su simulación.

2.2.2.2.1 Análisis de la propuesta AMENITIES

AMENITIES incluye la definición de un marco conceptual (Gea et al, 2003) que, a diferencia de otras propuestas, concede importancia a aspectos relativos al grupo y su comportamiento. Destacan en él la inclusión del concepto restricción (que a su vez se subdivide en ley y capacidad) que añade información adicional, con respecto a otras propuestas, acerca de la organización en la que se desarrolla en trabajo en grupo. Dentro del marco conceptual de AMENITIES se identifican cuatro modelos: El modelo de requisitos, el modelo formal permite llevar a cabo un análisis automatizado del sistema, el modelo cooperativo y el modelo de desarrollo software.

AMENITIES es una propuesta que pone énfasis en el modelado de la organización y de los planes. Para esta investigación, es importante resaltar que este marco conceptual no permite generar sistemas completamente ejecutables. Se clasifica entre las propuestas orientadas a desarrollar funcionalidades que mejoran la faceta de la colaboración y la cooperación en los sistemas interactivos.

2.2.2.3 Task-oriented and user-centred process model for developing interfaces for human-computer-human environments (TOUCHE)

TOUCHE (Penichet, 2007) es un enfoque metodológico que se basa en varios modelos para realizar el análisis de sistemas CSCW. Esta metodología permite abordar la etapa de análisis por medio de mecanismos para especificar la organización de los participantes de un sistema, los roles que desempeñan, la interacción de los usuarios con el sistema y la interacción entre los diferentes participantes a través del sistema, es decir, la interacción persona-computador-persona.

Touche posee una definición de modelo de proceso y una metodología para llevar a cabo cada etapa, en torno al desarrollo de interfaces de usuario para sistemas CSCW. Se basa en la ejecución iterativa de varias etapas de desarrollo, la recopilación de requisitos por medio de plantillas y un gran conjunto de modelos. Touche está centrado en las necesidades de los participantes como miembros de un grupo y considera las tareas que el participante tiene que realizar, principalmente las del grupo.



Las principales etapas de Touche son cuatro: Elicitación de requisitos, análisis, diseño e implementación. En realidad, corresponden a las etapas clásicas en desarrollo de software, pero con un énfasis en los aspectos característicos de los sistemas CSCW.

2.2.2.3.1 Análisis de la propuesta TOUCHE

Touche propone nuevos modelos y diagramas que permiten llevar a cabo el modelado tanto estático como de comportamiento que están asociados con los sistemas colaborativos, principalmente, para modelar las interacciones entre los usuarios. Estos modelos y diagramas permiten describir características tales como la colaboración entre los participantes del sistema o las interacciones persona-computador-persona, la organización de los participantes y los roles que desempeñan. De esta manera es posible desde la perspectiva de HCI construir interfaces para este tipo de sistemas.

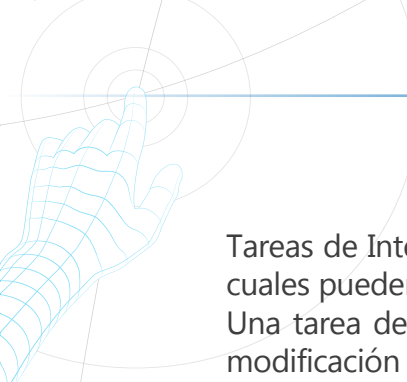
En TOUCHE la descripción de las tareas se hace mediante de la notación CTT (ConcurTaskTrees) (Paternò, 2004), a partir de una descomposición de tareas de manera jerárquica. Las tareas son identificadas y descritas a partir de la información de los requerimientos iniciales. Las tareas son documentadas mediante una plantilla que captura información relacionada con el trabajo en grupo (como tiempo, espacio, necesidades de comunicación, coordinación, colaboración, etc). A partir de las descripciones de las tareas se puede obtener el comportamiento del sistema. Este comportamiento se representa mediante el diagrama de co-interacción (CD) y el diagrama de tareas (TD). El diagrama de co-interacción tiene el propósito de modelar las interacciones que tienen lugar entre los participantes por medio del computador. Se observa que esta propuesta utiliza los casos de uso para modelar el nivel del sistema y en cuanto al diagrama de co-interacción, aunque ofrece información muy completa se puede observar que el elemento de modelado carece de expresividad.

2.2.3 Modelado y ejecución de la actividad en sistemas interactivos, interfaz de usuario

En esta sección se presentarán los principales estudios encontrados en la literatura relacionados con el modelado de la actividad en sistemas interactivos, y que se enfocan en el desarrollo de la **interfaz de usuario**. Luego de su descripción, se presenta un breve análisis para cada uno de ellos. El análisis de cada propuesta sirve como referencia para orientar la investigación y resulta útil para identificar las ventajas y desventajas de las propuestas existentes.

2.2.3.1 ConcurrTaskTrees (CTT)

La notación ConcurTaskTrees (CTT) (Paternò, 2004), (Paternò et al, 1997), toma como base el lenguaje LOTOS (Bolognesi et al, 1987) para definir modelos de tareas. CTT proporciona una representación gráfica sencilla de la estructura jerárquica de las tareas interactivas y usa operadores de LOTOS para expresar las relaciones temporales entre tareas de un mismo nivel de abstracción. Por medio de CTT es posible crear árboles de tareas con varios niveles de abstracción a partir de cinco tipos de tareas (Tareas de Usuario,



Tareas de Interacción, Tareas de Aplicación, Tareas Abstractas y Tareas Cooperativas), las cuales pueden relacionarse entre sí mediante su amplio juego de operadores temporales. Una tarea define cómo un usuario puede alcanzar una meta. Esta meta puede ser una modificación del estado del sistema o una consulta al mismo.

Las tareas en CTT se describen mediante una serie de atributos, dentro de los cuales se destacan los siguientes: nombre, tipo, objetos, importancia, prioridad, orden de acción, etc.

CTT usa la herramienta CTTE para editar el modelo de tareas utilizado para especificar tareas, roles y objetos, así como la jerarquía de tareas con operadores temporales.

En el caso de CTT una *tarea cooperativa* es entendida como una tarea que se descompone en subtareas a realizar por *roles* distintos. En este sentido, el modelo de tareas CTT está compuesto por un modelo de tareas para cada rol involucrado, más una parte cooperativa cuyo objetivo es indicar las relaciones entre las tareas desempeñadas por diferentes usuarios. La parte cooperativa se describe de forma similar a la interacción que puede realizar un solo usuario, con la única diferencia de que incluye tareas de cooperación (las tareas que implican acciones de dos o más usuarios) (Paternò et al, 1998).

2.2.3.1.1 Análisis de la propuesta CTT

CTT es una notación ampliamente utilizada en la comunidad HCI. En cuanto al modelado de la actividad, que es nuestro tema de interés, entre los tipos de tareas que se representan en CTT, destacan las tareas de tipo *interacción, aplicación y cooperación*. CTT tiene asociadas varias herramientas de edición para el modelado de tareas y de generación automática de interfaces de usuario como CTTE y TERESA (Berti et al, 2004). Aunque no tiene una metodología de desarrollo asociada, CTT es utilizada en distintas metodologías para el desarrollo de sistemas interactivos, como CIAM, TOUCHE, WISDOM, CIAF, que serán descritas más adelante. Debido a que la mayoría de propuestas que hacen referencia o utilizan CTT, la utilizan como un medio para la generación de la Interfaz de Usuario, se ha catalogado como una propuesta enfocada en el desarrollo de la Interfaz de Usuario para sistemas interactivos.

2.2.3.2 HAMSTERS

Hamsters (Human centered Assessment and Modeling to Support Task Engineering for Resilient Systems) (Barboni et al, 2010), es una notación para modelado de tareas inspirada principalmente en la notación CTT (ConcurTaskTrees) (Paternò, 2004). Realiza una extensión de esta notación e incluye clasificadores más detallados para el modelado de tareas.

HAMSTERS tiene asociado un ambiente de desarrollo que provee un conjunto de herramientas para edición y simulación de modelos de tareas. Además de esto, hace parte de una propuesta que se enfoca en integrar la ejecución de los modelos de tareas (centrados en el comportamiento del usuario) y los modelos del sistema (centrados en el comportamiento del sistema). Dicha propuesta expone una arquitectura para la

co-ejecución de los modelos mencionados (Figura 2.6). En esta arquitectura, la edición y simulación del modelo de tareas está soportada por el conjunto de herramientas que conforman el ambiente HAMSTERS, y la edición y simulación del modelo del sistema está soportada por la herramienta CASE PetShop, que soporta a su vez la notación ICO (Interactive Cooperative Objects) (Navarre, 2009).

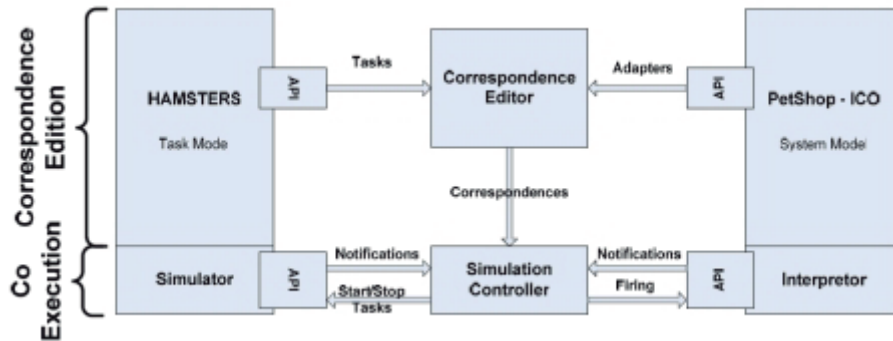


Figura 2.6. Arquitectura del Framework para la co-ejecución de modelos de tareas y de sistema (Barboni et al, 2010).

En esta arquitectura, la edición y simulación del modelo de tareas está soportada por el conjunto de herramientas que conforman el ambiente HAMSTERS, y la edición y simulación del modelo del sistema está soportada por la herramienta CASE PetShop, que soporta a su vez la notación ICO (Interactive Cooperative Objects) (Navarre et al, 2009).

2.2.3.2.1 Análisis de la propuesta HAMSTERS

La propuesta para el modelado de tareas HAMSTERS nos interesa principalmente porque se inspira en la notación CTT. Resulta interesante que extiende algunos de sus clasificadores, como "User Task" e "Interactive Task". Por otro lado, nos interesa analizar si lo que interesa modelar son las tareas de entrada o salida (Input Task, Output Task, Input Output Task), o más bien por ejemplo tareas de *feedback* (cada que el sistema hace algo y el usuario percibe el avance o el resultado de dicha actividad), de *awareness* (es la información que presenta el sistema acerca de lo que están haciendo los otros usuarios en tiempo real, ej: la imagen del lápiz de skype que indica que la otra persona está escribiendo), de seguridad, de error, de estado, etc.

Por otro lado, en la propuesta que integra el modelo de tareas con el modelo del sistema, el modelo de tareas se refiere a la especificación de tareas interactivas, es decir, esta especificación se ubica en el nivel de abstracción de sistema (según nuestra propuesta), al igual que el modelo de sistema, representado a través de la notación ICO. El ambiente de simulación interpreta el conjunto de entradas y salidas del modelo del sistema, como las funciones de activación y de representación que requiere el módulo de simulación.

La arquitectura del framework para la co-ejecución de los modelos de tareas y de sistema resulta muy completa para este estudio, sobre todo porque puede aportar la definición de

Por otro lado, en la propuesta que integra el modelo de tareas con el modelo del sistema, el modelo de tareas se refiere a la especificación de tareas interactivas, es decir, esta especificación se ubica en el nivel de abstracción de sistema (según nuestra propuesta), al igual que el modelo de sistema, representado a través de la notación ICO. El ambiente de simulación interpreta el conjunto de entradas y salidas del modelo del sistema, como las funciones de activación y de representación que requiere el módulo de simulación.

La arquitectura del framework para la co-ejecución de los modelos de tareas y de sistema resulta muy completa para este estudio, sobre todo porque puede aportar la definición de clasificadores en el nivel de abstracción de sistema y además la notación ICO puede tomarse como base en la definición de un lenguaje intermedio que represente la actividad entre los niveles de abstracción de Sistema y de Tecnología. HAMSTERS se ha catalogado en este análisis como una propuesta enfocada en el desarrollo de la interfaz de usuario para sistemas interactivos.

2.2.3.3 AMBOSS

AMBOSS es un ambiente para modelado de tareas propuesto por Giese (Giese et al, 2008), que se enfoca en la especificación de aspectos críticos de seguridad en los sistemas. Los modelos de tareas se complementan con información adicional que se refiere a aspectos relacionados con el tiempo, información espacial y comunicación. Cada modelo de tareas consta de una cantidad específica de tareas. Una tarea es una caracterización de una determinada actividad y se compone de varias subtareas. La figura 2.7 describe los atributos que se asignan a cada tarea y cómo se representa el paso de mensajes entre tareas.

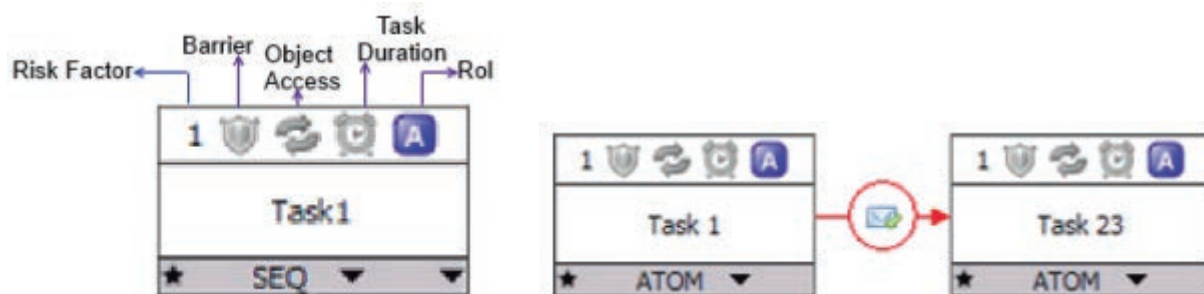
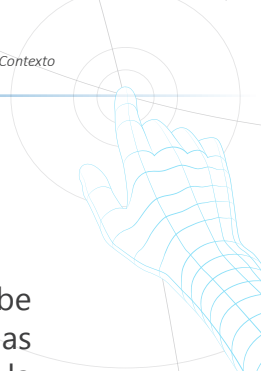


Figura 2.7. Representación de una tarea, y del paso de mensajes entre tareas en AMBOSS

Cada tarea en el modelo tiene varios atributos, los cuales describen el entorno y el comportamiento de la tarea. Esta propuesta también permite el paso de mensajes entre tareas. Los mensajes tienen un conjunto de atributos que permiten especificar de forma completa cada mensaje (Arteaga et al, 2015).



2.2.3.3.1 Análisis de la propuesta AMBOSS

Al igual que la mayoría de las notaciones para modelado de tareas, AMBOSS describe una jerarquía para representar tareas de alto nivel que son descritas por subtareas, las cuales tienen relaciones entre sí. Se destaca la cantidad de atributos definidos para la especificación de las tareas; son necesarios debido a la naturaleza de las aplicaciones que se modelan con esta notación. Un aspecto importante para analizar es el nivel de expresividad en el uso de colores para diferenciar distintos valores en los atributos de las tareas. Aunque la propuesta AMBOSS no se enfoca en el desarrollo de la interfaz de usuario, se tiene en cuenta en esta sección por la representación que propone para el modelado de tareas.

2.2.3.4 UCD with activity modeling

El modelado de la actividad es una aproximación sistemática, propuesta por Constantine (Constantine, 2006) que se encuentra fundamentada teóricamente sobre la Teoría de la Actividad y metodológicamente sobre el Diseño Centrado en el Uso. Lo que se pretende es integrar la Teoría de la Actividad con el Diseño centrado en el uso para proveer una organización más fuerte y coherente de dichas propuestas. La notación que se propone es una extensión de la notación de UML. Los modelos que propone Constantine están orientados a maximizar la utilidad y eficiencia en la representación del contexto de la actividad para propósitos del diseño de la interacción en lugar de la ingeniería de software. La Figura 2.8 muestra los símbolos de los elementos notacionales establecidos para el Modelado de la Actividad.









Symbol	Name	Description
	actor, user actor	activity participant interacting with the system of reference
	role, user role	relationship between an actor and the system of reference
	system actor	non-human system (software or hardware) interacting with the system of reference
	player*	activity participant not interacting with the system of reference (but often an actor with other systems)
	artifact, tool*	any artifact employed within an activity
	activity*	collection of actions or tasks undertaken for some purpose
	task, task case	action by an actor in interaction with the system of reference for some goal within an activity
	action*	action by a player for some goal within an activity

Figura 2.8. Notación de diseño centrado en el uso, extendida, para el modelado de la actividad (Constantine, 2006).

La Figura 2.9 muestra una vista del Diseño Centrado en el Uso con el Modelado de la Actividad. La vista se divide en Modelado de la Actividad y Modelado de la Solución. El primero, expresa el problema desde una perspectiva centrada en la actividad, mientras que el segundo expresa el diseño que sigue desde la expresión del problema.

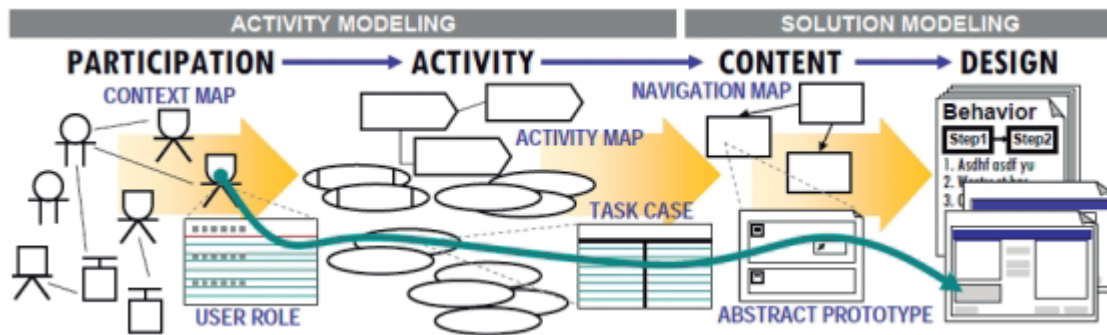


Figura 2.9. Vista lógica del diseño centrado en el uso con el modelado de la actividad (Constantine, 2006)

El modelo de la actividad en sí mismo incluye tres partes: Un Mapa de Actividad que identifica actividades relevantes y sus relaciones, una colección de Perfiles de Actividad que describe los aspectos sobresalientes de las actividades relevantes, y un Mapa de Participación, que muestra la participación de los actores con el sistema, con otros artefactos y con otros participantes.

2.2.3.4.1 Análisis de la propuesta UCD with activity modeling

Es interesante la clasificación de Roles que realiza Constantine en su propuesta. Para la definición de la Taxonomía de la Actividad se analiza dicha clasificación. Igualmente, las relaciones que se establecen entre las actividades. Se ha clasificado esta propuesta en el nivel de abstracción de Sistema, debido a que los modelos sólo contemplan elementos que relacionan los actores y roles con el sistema interactivo. Se destaca la utilidad de la notación para el Modelado de la Actividad, para la definición de clasificadores de la TxA en la columna del "Quién".

2.2.3.5 Propuesta de Modelado de Trætterberg

Trætterberg (Trætterberg, 2002), propone un método basado en modelos para el desarrollo de la interfaz de usuario. En su propuesta aborda dos enfoques, por un lado el enfoque formal, analítico, top-down y por el otro lado el enfoque informal, sintético, bottom-up, para definir un marco para la clasificación de las representaciones a emplear en el desarrollo de la interfaz de usuario.

En el desarrollo de sus lenguajes (TaskMODL y DiaMODL) utiliza un framework para la clasificación desde tres perspectivas de diseño de la interfaz de usuario: las tareas, el diálogo abstracto y la interacción concreta. TaskMODL representa el modelado de las tareas combinando características de los lenguajes de modelado Workflow y de

modelado de tareas. Este lenguaje se integra con el lenguaje de modelado del dominio de la aplicación (RML) que permite modelar la parte estática del dominio del problema. DiaMODL representa el modelado del diálogo mediante el uso de componentes de interacción (en inglés, interactores). Este lenguaje permite modelar componentes de interacción compuestos y parametrizables, y presenta una estructura de diagrama de estados. El modelo de interacción concreta se basa en el lenguaje de modelado del diálogo, y en él ya se incluyen elementos estándar de interfaces basadas en ventanas, así como relaciones de composición entre dichos elementos.

2.2.3.5.1 Análisis de la propuesta de Trætteberg

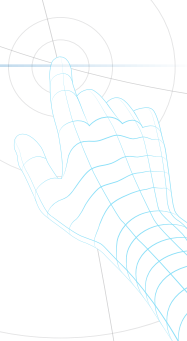
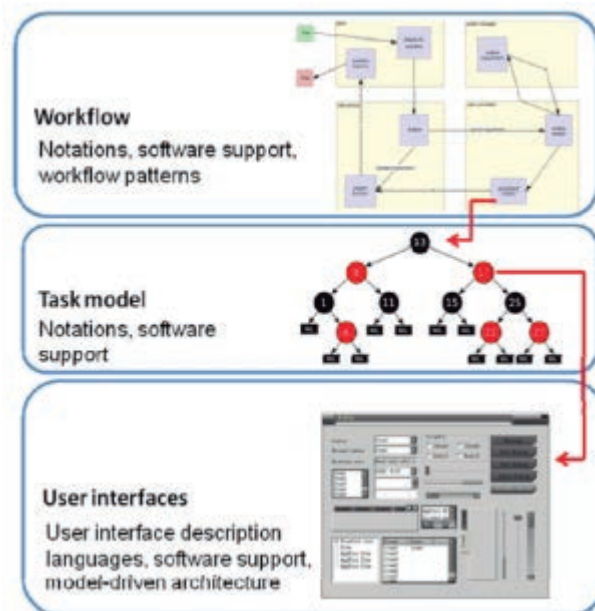
La propuesta de Trætteberg es interesante para este trabajo, porque a diferencia de la mayoría, contempla elementos de modelado de la actividad que se clasifican en el nivel de abstracción de negocio. Dichos elementos se enfocan en el modelado del "Cómo" y del "Quién".

2.2.3.6 UI to Workflow IS

Guerrero (Guerrero, 2010), propone un método para el desarrollo de interfaces de usuario de un sistema de información workflow. El método incluye un lenguaje para especificar la Interfaz de Usuario, una metodología para estructurar el uso de dichos modelos y un software de soporte. Se descompone entonces un workflow en procesos que a su vez se descomponen en tareas. Cada tarea da lugar a un modelo de tareas cuya estructura, orden y conexión con el modelo de dominio permite una generación semi-automática de las interfaces de usuario correspondientes (Figura 2.10).

La notación utilizada en los modelos de esta propuesta, retoma elementos de las redes de petrí y de los modelos de tareas de CTT.

Figura 2.10. Vista general de la propuesta metodológica para desarrollar UIs para sistemas de información basados en workflow (Guerrero, 2010)



2.2.3.6.1 Análisis de la propuesta IU to Workflow IS

El nivel de workflow que tiene en cuenta esta propuesta provee elementos de modelado, por ejemplo los "Organization Units" que permiten tener una vista particionada de las tareas en el nivel de abstracción de Negocio, a diferencia de la mayoría de propuestas centradas en el desarrollo de interfaces de usuario. El hecho de retomar la notación de redes de petrí, permite contar con una vista de concurrencia en las tareas. Es interesante cómo en esta propuesta metodológica se introducen elementos de modelado de Workflow relacionados con el tiempo (cuándo se ejecuta la actividad), como el concepto de "Agenda". Este elemento podría integrarse a la definición de la TxA para tener una especificación más completa de la Actividad en los sistemas interactivos.

2.2.3.7 Interface development environment within OASIS (IDEAS)

IDEAS (Lozano, 2001) es una metodología completa para el desarrollo de interfaces de usuario interactivas. IDEAS se centra en el desarrollo de un enfoque integrado en la Ingeniería del Software, y la elaboración de modelos por medio de la notación OASIS (Letelier et al, 1998). IDEAS presenta el diseño de la interfaz abstracta hecha a mano y más orientado a los datos que a las tareas. En su fase de diseño propone un modelo de diálogo que describe la interfaz gráfica a partir de cuatro diagramas: diagrama de interacción de diálogos, diagrama de estados internos, diagrama de especificación de componentes y tabla de definición de componentes. Finalmente, propone el uso de contratos para relacionar Objetos de Interacción Abstractos (AIOs) con objetos del dominio.

2.2.3.7.1 Análisis de la propuesta IDEAS

En la propuesta de IDEAS se observa que el modelo de tareas se expresa en modo texto a través de plantillas. Se utilizan los diagramas de casos de uso del lenguaje UML y también se describe la estructura sintáctica de la interacción usuario-sistema a través de diagramas de Interacción de Diálogos. Adicionalmente, se observa que en el diagrama de Casos de Uso Refinado se mezclan las vistas y los niveles de abstracción, ya que se muestran en un mismo diagrama los casos de uso de sistema, reglas de negocio y clases. Así mismo, en el Diagrama de Secuencia I se observa que se mezclan niveles de granularidad.

IDEAS resulta interesante en el sentido de que los diagramas que soportan el diseño de la interfaz de usuario y la definición de los "contratos", pueden aportar clasificadores en la estructura taxonómica de la TxA. Para este estudio, se clasifica IDEAS como una propuesta enfocada en el desarrollo de la Interfaz de Usuario.

2.2.3.8 MultiPath

MultiPath (Limbourg y Vanderdonckt, 2004b) es un enfoque de desarrollo de la interfaz de usuario sobre la base del proyecto CAMALEÓN², que se centra en los

² <http://giove.isti.cnr.it/projects/cameleon.html>



aspectos técnicos de la adaptación a múltiples contextos y dispositivos. Multipath se basa en el desarrollo transformacional para hacer un refinamiento progresivo de modelos abstractos en modelos concretos, hasta lograr obtener el código del programa. Este enfoque se realiza mediante varias vías de desarrollo (por ejemplo, ingeniería hacia adelante, ingeniería inversa, adaptación al contexto de uso) a fin de lograr el desarrollo mediante múltiples rutas. Para tal fin, MultiPath utiliza el lenguaje usiXML (Limbourg et al, 2004) que define los distintos puntos de vista que pueden ser mantenidos en una interfaz de usuario de un sistema interactivo. Los distintos puntos de vista se estructuran jerárquicamente según su nivel de abstracción para describir las tareas del usuario, las clases de objetos, los aspectos de la presentación y del comportamiento de las interfaces de usuario, el contexto de uso, y un conjunto de mapping entre estas representaciones. MultiPath cuenta con una gran variedad de heurísticas de transformación para expresar múltiples caminos de desarrollo.

2.2.3.8.1 Análisis de la propuesta MultiPath

La propuesta de Limbourg es importante para este trabajo porque el hecho de que utilice el lenguaje usiXML la convierte en base de otras propuestas interesantes como de Guerrero (Guerrero, 2010), la de Giraldo (Giraldo, 2010), entre otras. En general, Limbourg es un gran referente para la mayoría de las propuestas que enfocadas en el desarrollo de la interfaz de usuario. Se destaca también que Limbourg utiliza la notación CTT para el modelado de tareas.

2.2.3.9 DYGIMES

Dygimes (Luyten, 2004) es un framework para generación dinámica de interfaces de usuario para sistemas embebidos y dispositivos para computación móvil. Utiliza descripciones de interfaz de usuario de alto nivel basadas en XML, en combinación con una especificación de tarea, una especificación de interacción y restricciones de diseño espacial.

2.2.3.9.1 Análisis de la propuesta DYGIMES

Al igual que Limbourg, la propuesta de Luyten se centra en el desarrollo de la interfaz de usuario y utiliza la notación CTT para expresar los modelos de tareas. Se destaca que propone un algoritmo para la generación de interfaces de usuario, a partir de los modelos de tareas.

2.2.3.10 MPIu+a

MPIu+a (Modelo de Proceso de la Ingeniería de la Usabilidad y la Accesibilidad) (Granollers, 2004) es un proceso que involucra el prototipado y la evaluación en todas las disciplinas del proceso de tal forma que involucra al usuario como un elemento central en el desarrollo. La figura 2.11 muestra el mapa de ruta propuesto por MPIu+a.

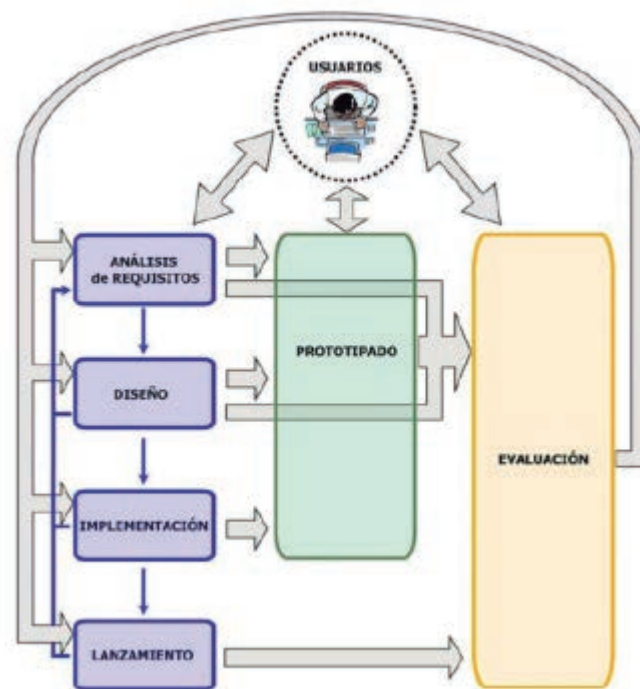


Figura 2.11. Mapa de ruta propuesto en MPIu+a (Granollers, 2004)

2.2.3.10.1 Análisis de la propuesta MPIu+a

MPIu+a utiliza HTA (Hierarchical Task Analysis) (Annett et al, 1967) para modelar la actividad, que de acuerdo con la propuesta de clasificación, se ubica en el nivel de abstracción de negocio porque sólo modela las tareas de las personas y no permite modelar las tareas del sistema interactivo. También utiliza CTT (ConcurTaskTrees) (Paternò, 2004) que permite modelar tanto las tareas de las personas como las tareas del sistema interactivo. Sin embargo, la metodología no hace la distinción entre los niveles de abstracción Negocio y Sistema. Se destaca que MPIu+a organiza y describe un catálogo muy completo de pruebas de usabilidad.

2.2.3.11 Collaborative interaction application framework (CIAF)

CIAF (Collaborative Interaction Application Framework) (Giraldo, 2010) es un marco de desarrollo para sistemas interactivos colaborativos, que integra el desarrollo de la Funcionalidad, de la Colaboración y de la Interfaz de Usuario. Particularmente, para esta última, propone una metodología llamada TD-MBUID (*Task & Data-Model Based User Interface Development*) que se centra en combinar el diseño de las interfaces basadas en modelos de datos y de tareas.

La especificación y uso del modelo de tareas y de interacción se ha hecho de forma general para que pueda desarrollarse la interfaz de usuario con distintas propuestas para el modelado de la actividad. Como requisito para que se pueda realizar la integración, se indica que, dichas propuestas deben soportar las categorías de la

clasificación del modelado de la actividad referenciadas en este trabajo: Aspecto o Faceta, Nivel de Abstracción, Nivel de Granularidad y Tipo de Actividad. TD-MBUID realiza la integración entre casos de uso de UML, modelado de tareas de CIAN (Molina, 2006) y modelado de tareas de CTT (Paternò, 2004). La figura 2.12 muestra el mapa de ruta para el desarrollo de la interfaz de usuario propuesto por TD-MBUID.

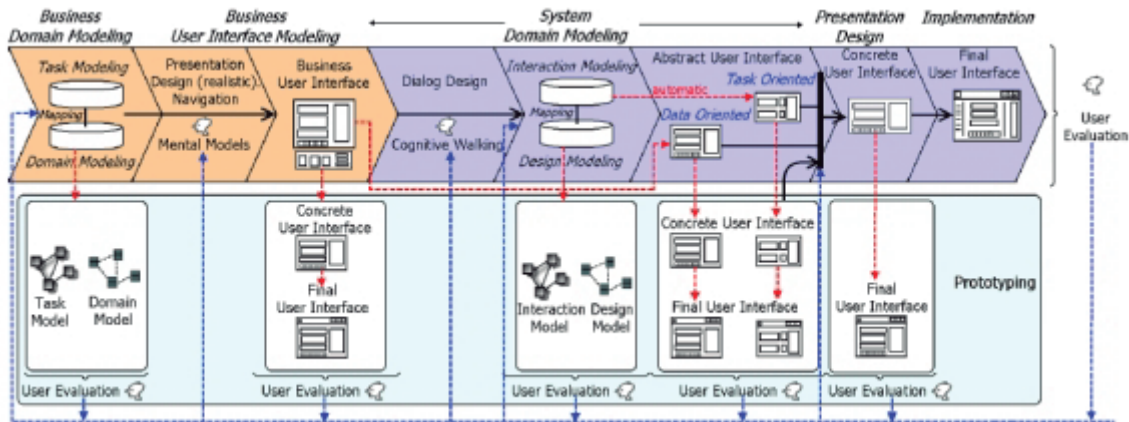


Figura 2.12. Mapa de ruta para el desarrollo de la interfaz de usuario en TD-MBUID (Giraldo, 2010)

2.2.3.11.1 Análisis de la propuesta CIAF

CIAF es una de las propuestas que más aporta a este trabajo, porque integra gran parte de los conceptos y artefactos contenidos en las demás propuestas analizadas en esta sección. Adicionalmente, incluye la Clasificación de la Actividad a partir de la cual se define la Taxonomía de la Actividad en este trabajo.

El conjunto de propuestas analizadas proporciona información acerca de los elementos que cada una de ellas podría aportar en la definición de la Taxonomía de la Actividad, definida como uno de los componentes principales de este trabajo.

La información recopilada se resume en elementos conceptuales, expresados a través de plantillas, diagramas, modelos y metamodelos, que conforman las propuestas analizadas.

2.2.4 Taxonomías en ingeniería de software y HCI

La revisión de la literatura relacionada con Taxonomías en el ámbito de la Ingeniería de Software (IS) y de HCI permite recopilar información acerca de las principales características de dichas taxonomías. Se han definido un conjunto de criterios que permiten analizar el enfoque de cada una de ellas y también el conjunto de clasificadores que las conforman.

Las Tablas 2.2 y 2.3 contienen la información detallada del análisis realizado, de acuerdo con: *Área de enfoque, Qué Clasifica, Conjunto de Clasificadores, Clasificadores por Nivel de Abstracción, ¿Define Proceso de Clasificación?, ¿Define Estructura taxonómica?, ¿Define Reglas de la Taxonomía?*.

Tabla 2.2(a). Descripción de Taxonomías en Ingeniería de Software y HCI

Propuesta	Área	Qué clasifica	Clasificadores
REST Taxonomy (Unterkalmsteiner et al, 2014)	IS	Métodos para alinear la Ing. De Requisitos (IR) con las Pruebas de Software (PS)	<ul style="list-style-type: none"> - Complejidad: (Número de métodos, Ramificaciones, Métodos intermedios, Tipo de links “con IR”, Tipo de links “entre fases”, Tipo de links “con PS”). - Enfoque / alcance: (Proporción IR:PS, Alcance).
A taxonomy of software development Methods (Blum, 1994)	IS	Métodos usados en el desarrollo de software	<ul style="list-style-type: none"> - Conceptual (Orientado al problema, Orientado al producto). - Formal (Orientado al problema, Orientado al producto).
Zachman Taxonomy (Sowa et al, 1992)	IS	Información contenida en los modelos que representan las arquitecturas empresariales.	<ul style="list-style-type: none"> - Datos (Qué) - Proceso (Cómo) - Red (Dónde) - Personas (Quién) - Tiempo (Cuándo) - Motivación (Por Qué)
Software Engineering Taxonomy (Stoll et al, 2009)	IS	Artefactos utilizados en la Ingeniería de Software según el SWEBOK {Abran, 2004 #230}.	<ul style="list-style-type: none"> - Datos (Qué) - Proceso (Cómo) - Red (Dónde) - Personas (Quién) - Tiempo (Cuándo) - Motivación (Por Qué)
Software Development Risk Taxonomy (Carr et al, 1993)	IS	Riesgos asociados con el desarrollo de un proyecto que involucra software.	<ul style="list-style-type: none"> - Clase: (Ingeniería de Producto, Ambiente de Desarrollo, Restricciones de Programa). - Elemento: (Requisitos,..., Especialidades en Ingeniería, Procesos de desarrollo,..., Ambiente de trabajo, Recursos,..., Interfaces de programa). - Atributo: (Estabilidad,..., Escala, Formalidad,..., Control de producto, Planeación,..., Facilidades).
Taxonomy for technology adoption (Seneler et al, 2008)	HCI	Los factores que afectan la aceptación de tecnología y la intención del usuario en utilizar un producto.	<ul style="list-style-type: none"> - Tarea - Producto: (No-interfaz, Interfaz:, Alta interacción) - Contenido de información - Social: (Contexto del sistema, Usuario y Estados mentales y emocionales) - Intermediario
Taxonomy of HCI (Lessiter et al, 2014)	HCI	Un rango de variables que impactan sobre los tipos de posibles interacciones humano computador, incluyendo interacciones simbióticas.	<ul style="list-style-type: none"> - Componentes: (Usuario, Agente autónomo sensible/computador, display de contenido/datos) - Interacciones activas y pasivas - Tipos de respuestas de usuarios: (Explícitas, implícitas)
Taxonomy of workflow management systems for grid computing (Yu et al, 2005)	IS	Propuestas para construir y ejecutar workflows sobre grids.	<ul style="list-style-type: none"> - Diseño del workflow: (Estructura, Modelo/especificación, Sistema de Composición, Restricciones de calidad). - Planeación del workflow: (Arquitectura, Toma de decisiones, esquema de planeación, estrategias, estimación de rendimiento). - Recuperación de la información: (Estática, histórica, dinámica) - Tolerancia a fallos: (Nivel de tareas, nivel de workflow) - Movimiento de datos: (dirigido por usuario automático)

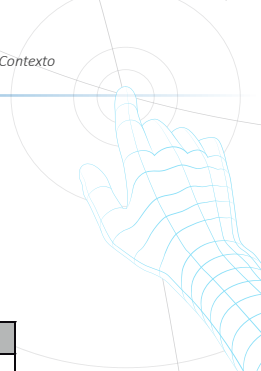


Tabla 2.2(b). Descripción de Taxonomías en Ingeniería de Software y HCI

Propuesta	Área	Qué clasifica	Clasificadores
Taxonomy for Visual Analytics (von Landesberger et al, 2014)	HCI	Interacciones de los usuarios en aplicaciones interactivas de análisis visual	<ul style="list-style-type: none"> - Visualización de la información: (Cambios en los datos, Cambios en la visualización) - Razonamiento: (Cambios en los datos, Cambios en el razonamiento) - Procesamiento de datos: (Cambios en los datos, Cambios en el procesamiento)
Taxonomía de aplicaciones móviles (Nickerson et al, 2009)	IS-HCI	Características de aplicaciones móviles	<ul style="list-style-type: none"> - Temporal: (Síncrono, Asíncrono) - Comunicación: (Informacional, Reporte, Interaccional) - Transacción: (Transaccional, No-Transaccional) - Público: (Público, Privado) - Multiplicidad: (Individual, Grupo) - Ubicación: (Basado en localización, No basado en localización) - Identidad: (Basado en la identidad, No basado en la identidad)
A taxonomy of Gestures in Human Computer Interaction (Karam et al, 2005)	HCI	Interacciones humano computador basadas en gestos	<ul style="list-style-type: none"> - Dominio de la aplicación: (Escritorio, Móvil y pervasivo, Ubicua, Virtual y RA, Juegos, Tecnologías adaptativas, Comunicación, Entretenimiento, Interacciones multimodales) - Tecnologías habilitadas: (Perceptiva, No-Perceptiva) - Respuesta del sistema: (Visual, Audio, Comandos) - Estilos de gestos: (Deíctico, Gesticulación, Manipulación, Semáforos, Lenguaje de Señas)
Taxonomy of Human Interruption (McFarlane, 1998)	HCI	La interrupción humana en HCI	<ul style="list-style-type: none"> - Fuente de la interrupción - Características de la persona que recibe la interrupción - Método de coordinación - Significado de la interrupción - Método de expresión - Canal de transporte - Actividad humana cambiada por la interrupción - Efecto de la interrupción
Level of automation taxonomy (Save et al, 2012)	IS-HCI	Tipos de soporte a la automatización en la Administración de Tráfico Aéreo (ATM)	<ul style="list-style-type: none"> - Adquisición de la información: (Manual, Soportado por artefactos, Por niveles de automatización) - Análisis de la información: (Basado en el trabajo de la memoria, Soportado por artefactos, Por niveles de automatización) - Decisión y selección de la acción: (Decisión humana, Soportado por artefactos, Soporte automático, Soporte automático rígido, Por niveles de automatización) - Implementación de la acción: (Control y acción manual, Soportada por artefactos, Soporte paso a paso, Bajo nivel de soporte, Alto nivel de soporte, Por niveles de automatización)
Taxonomía para sistemas interactivos colaborativos (Giraldo, 2010)	IS-HCI	Información expresada en los modelos que especifican los sistemas que dan soporte al trabajo colaborativo	<ul style="list-style-type: none"> - Datos (Qué) - Proceso (Cómo) - Red (Dónde) - Personas (Quién) - Tiempo (Cuándo) - Motivación (Por Qué)

Tabla 2.3. Características Generales de Taxonomías en Ingeniería de Software y HCI

Propuesta	Clasificadores por nivel de abstracción	Proceso de clasificación	Estructura taxonómica	Reglas de la taxonomía
REST Taxonomy (Unterkalmsteiner et al, 2014)	--	Si	Si	No
A taxonomy of software development Methods (Blum, 1994)	Conceptual (sistema) Formal (tecnología)	Si	Si	No
Zachman Taxonomy (Sowa et al, 1992)	Alcance Modelo de Negocio Modelo de Sistema Modelo de Tecnología Componentes	No	Si	Si
Software Engineering Taxonomy (Stoll et al, 2009)	Alcance Modelo de Negocio Modelo de Sistema Modelo de Tecnología Componentes	Si	Si	Si
Software Development Risk Taxonomy (Carr et al, 1993)	--	Si	Si	No
Taxonomy for technology adoption (Seneler et al, 2008)	--	No	Si	No
Taxonomy of HCI (Lessiter et al, 2014)	--	No	No	No
Taxonomy of workflow management systems for grid computing (Yu et al, 2005)	- Diseño del workflow - Planeación del workflow	No	Si	No
Taxonomy for Visual Analytics (von Landesberger et al, 2014)	--	Si	Si	No
Taxonomía de aplicaciones móviles (Nickerson et al, 2009)	--	No	Si	No
A taxonomy of Gestures in Human Computer Interaction (Karam et al, 2005)	--	No	Si	No
Taxonomy of Human Interruption (McFarlane, 1998)	--	Si	No	No
Level of automation taxonomy (Save et al, 2012)	--	No	Si	No
Taxonomía para sistemas interactivos colaborativos (Giraldo, 2010)	Modelo de Negocio Modelo de Sistema Modelo de Tecnología	Si	Si	Si



Es posible observar que sólo las propuestas basadas en el Framework de Zachman (Zachman, 1987), a saber, Software Engineering Taxonomy (Stoll et al, 2009) y Taxonomía para Sistemas Interactivos Colaborativos (Giraldo, 2010), definen las características generales "*Proceso de Clasificación*", "*Estructura Taxonómica*" y "*Reglas*". También definen sus clasificadores de acuerdo a la separación en niveles de abstracción. Adicionalmente, la mayoría de taxonomías se utilizan para comparar, identificar ventajas y carencias relacionadas con las características de propuestas que encajan en el conjunto de clasificadores definidos para dichas taxonomías.

Se observa también que la mayoría de propuestas se enfocan en un interés distinto a la clasificación de elementos relacionados con la labor o actividad y la mayoría de propuestas que se enfocan en el área de ingeniería de software se interesan por clasificar aspectos relacionados al proceso de desarrollo como tal y no al producto software, como es el caso de las siguientes Taxonomías: "REST" (Unterkalmsteiner et al, 2014), "Taxonomy of Software Development Methods" (Blum, 1994), "Software Development Risk Taxonomy" (Carr et al, 1993), y "Taxonomy of Workflow Management Systems for Grid Computing" (Yu et al, 2005). Particularmente, resulta interesante la clasificación de los artefactos propuestos por el cuerpo de conocimiento en Ingeniería de Software SWEBOK (Abran, 2004), de acuerdo a la taxonomía de Zachman (Sowa y Zachman, 1992).

Se observa que en la columna "Who" se clasifican los "Casos de Uso de Sistema", a pesar de que este artefacto está orientado a especificar la funcionalidad del sistema, es decir, lo que se refiere a la columna "How".

Por otra parte, las taxonomías enfocadas en el área de HCI definen clasificadores orientados al producto, considerando especificaciones en cuanto al factor humano y a la interacción. Estos clasificadores se refieren netamente a especificaciones de la actividad - labor en los sistemas interactivos.

En este capítulo se ha realizado un análisis profundo del estado del arte que soporta esta investigación. En este estudio encontramos que hasta ahora no se ha hecho un análisis que nos lleve a entender mejor la actividad, desde diferentes puntos de vista. Nuestro interés es integrar, de una manera adecuada, el desglose de un conjunto de piezas de información, que proviene de las contribuciones de un conjunto de expertos, para obtener artefactos que especifiquen la interfaz de usuario en un sistema interactivo.

En este sentido, se hace necesario plantear un enfoque taxonómico. Una taxonomía que no sólo clasifique la labor, sino también las demás dimensiones necesarias para interrelacionar las distintas representaciones de un sistema interactivo.

En el capítulo 3 se presenta la definición de la Taxonomía de la Actividad. Se describe en detalle el mapa de ruta que se siguió para tal fin.



TAXONOMÍA DE LA ACTIVIDAD PARA SISTEMAS INTERACTIVOS

CAPÍTULO TERCERO

3. TAXONOMÍA DE LA ACTIVIDAD PARA SISTEMAS INTERACTIVOS

3.1 TEORÍA DE LA ACTIVIDAD COMO BASE PARA LA DEFINICIÓN DE LA TXA

3.2 CLASIFICACIÓN DE LA ACTIVIDAD COMO BASE PARA LA DEFINICIÓN DE LA TXA

3.2.1 Aspecto o faceta

3.2.2 Nivel de abstracción

3.2.3 Nivel de granularidad

3.2.4 Tipo de actividad

3.2.5 Razones para elegir la clasificación del modelado de la actividad

3.3 FRAMEWORK DE ZACHMAN COMO BASE PARA LA DEFINICIÓN DE LA TXA

3.4 MAPA DE RUTA PARA LA DEFINICIÓN DE LA TXA

3.4.1 Clasificación de propuestas mediante el uso de la clasificación de la actividad

3.4.2 Integración de propuestas para la definición del metamodelo base

3.4.3 Clasificación de propuestas según la taxonomía de Zachman

3.4.4 Clasificadores complementarios para cada celda de la TxA



CAPÍTULO TERCERO

3. TAXONOMÍA DE LA ACTIVIDAD PARA SISTEMAS INTERACTIVOS

Luego de entender la motivación de esta investigación, que se trata de mejorar la expresividad, la integración y contribución de información a través de procesos de desarrollo de sistemas interactivos, se concluye que es necesaria la definición de una taxonomía apoyada por un buen marco de referencia y que permite entender cómo crear notaciones, metodologías y herramientas. Todo esto dentro de un road-map, que es una interpretación del marco de desarrollo CIAF (Giraldo, 2010).

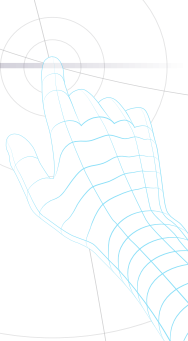
Las taxonomías de objetos y conceptos se han convertido sin duda en una herramienta científica básica, como se resume en (Unterkalmsteiner et al, 2014). Los sistemas de clasificación permiten la visualización de la teoría de una manera útil y servir, de forma similar a las teorías, como conductores para la investigación (Kwasnik, 1992). Particularmente, en el ámbito de la Ingeniería de Software, las taxonomías se convierten entonces en un elemento esencial para documentar teorías que acumulan el conocimiento (Sjoberg et al, 2007).

La Taxonomía de la Actividad (TxA) que se define en esta tesis se ha especificado en base al análisis de un conjunto de teorías, propuestas metodológicas, taxonomías y marcos de desarrollo, los cuales se describieron en el capítulo 2. Marco de Referencia. Se escoge la abreviatura TxA para evitar confusiones con la TA que es normalmente como se hace referencia a la Teoría de la Actividad.

Entre los fundamentos más importantes para el desarrollo de esta propuesta se encuentran la Teoría de la Actividad (Engeström, 1987), la Clasificación del Modelado de la Actividad (Giraldo, 2010), el Framework de Zachman (Sowa y Zachman, 1992) y el conjunto de propuestas para el modelado de la actividad que se describieron en la sección 2.2. "Antecedentes".

3.1 TEORÍA DE LA ACTIVIDAD COMO BASE PARA LA DEFINICIÓN DE LA TXA (Villegas et al,2016b)

El análisis de la Teoría de la Actividad (TA) y su integración con aspectos del desarrollo de sistemas interactivos es absolutamente necesaria en este trabajo, no sólo porque dicha teoría se ha definido como un paradigma psicológico (Georg et al, 2015), sino



también porque hasta el día de hoy se resalta su aplicación en la Ingeniería de Software (Georg, 2011), y en la Interacción Humano Computador (Nardi, 1996).

Específicamente, en el levantamiento de requisitos de sistemas interactivos complejos, la TA se ha definido como una herramienta potencial, principalmente, porque hace énfasis en el contexto social en el cual la actividad humana toma lugar (Georg et al, 2013).

En la literatura se observan diversos trabajos (Georg et al, 2015), (Georg et al, 2013), (Souza, 2003), (Neto et al, 2005b), en los que se propone integrar la TA con notaciones, lenguajes y metodologías para el levantamiento de requisitos de sistemas interactivos. Generalmente, estas especificaciones de requisitos se basan en plantillas que clasifican la actividad de una manera muy general sin permitir a los desarrolladores capturar información relevante como por ejemplo feedback, awareness, niveles de abstracción, tipos de función, etc. De esta forma es difícil para los desarrolladores tomar conciencia de toda la información que requiere ser capturada en dichas especificaciones. Estas deficiencias metodológicas generan un impacto negativo sobre el entendimiento y la percepción que el usuario final tiene sobre el sistema interactivo que se desarrolla. Lo que se busca es que el desarrollador comprenda perfectamente cuál es el impacto de una mala clasificación de la actividad en la adquisición de información, la aceptación del producto, en términos generales la calidad vista como el cumplimiento de las características del producto respecto de las necesidades de los usuarios.

La Taxonomía de la Actividad (TxA) se toma entonces como una herramienta que permite principalmente identificar elementos de la actividad (clasificadores) que la mayoría de los ingenieros de desarrollo de software no han considerado hasta el día de hoy y que deberían considerar si se tienen en mente atributos como la colaboración, la interfaz de usuario, la usabilidad, la televisión interactiva.

3.2 CLASIFICACIÓN DE LA ACTIVIDAD COMO BASE PARA LA DEFINICIÓN DE LA TXA

El concepto de modelado de la actividad se entiende como la práctica que se centra en la comprensión y representación de la naturaleza del trabajo con el fin de establecer una base para el diseño de un sistema que lo soporte (Iqbal et al, 2005). Se usa el concepto de Clasificación del Modelado de la Actividad, como la acción de clasificar los distintos tipos de elementos del modelado de la actividad, por medio de categorías necesarias para la localización de elementos de modelado específicos.

La Clasificación del Modelado de la Actividad brinda a los ingenieros, que pretenden integrar componentes metodológicos, la posibilidad de aplicar un marco de comparación de los elementos asociados al modelo de la actividad. Por ejemplo, en la literatura, el modelado de la funcionalidad por medio de casos de uso es considerado como distinto al modelado de tareas, sin embargo, ambos modelados encajan dentro

de esta clasificación, por ende pueden estar relacionados. Esta clasificación se centra sólo en las propuestas metodológicas que tienen como punto de partida, o eje conductor, el modelado de la actividad.

La actividad se toma casi siempre como un patrón que permite interrelacionar las distintas variables que dirigen el diseño. Por ejemplo, en la teoría de la actividad se relaciona la actividad con los objetivos, los usuarios, los recursos, etc (Engeström, 1987). A continuación se presentan las categorías a tener en cuenta cuando se define un modelo de la actividad.

3.2.1 Aspecto o faceta

La clasificación por aspectos, facetas o cualidades es la principal dimensión que se considera en la Clasificación de la Actividad. Se tiene que el modelo completo de la actividad de un sistema (Figura 3.1, izquierda) puede ser clasificado y representado por una o más capas que posean fracciones del modelado de la actividad para un aspecto en particular (Figura 3.1, derecha).

Cuando un sistema está definido de acuerdo a una serie de facetas el modelado de la actividad debe estar capacitado para representar las actividades que están relacionadas con dichas facetas. La división de la actividad se hace para entender el modo en el que la actividad puede ser representada por (Limbourg, 2004): a. La Ingeniería del Software, b. La Planificación y asignación de la actividad, c. Los Estudios Etnográficos de la Actividad, d. La Psicología Cognitiva, e. La Seguridad, El Rendimiento, etc.

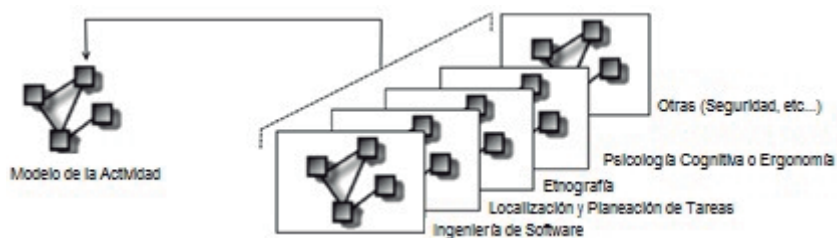


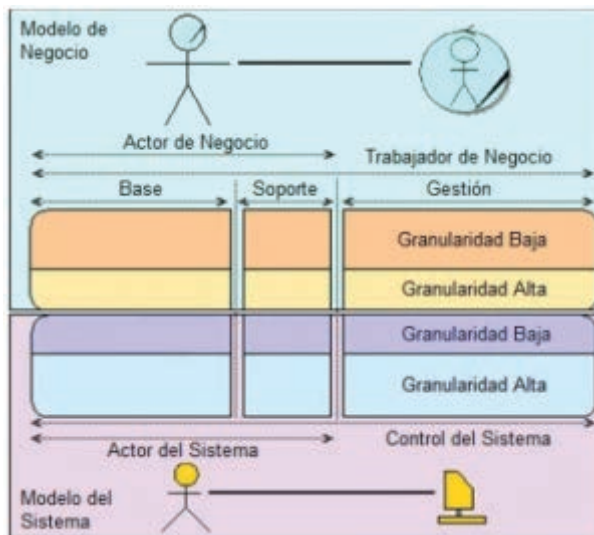
Figura 3.1. Clasificación de la actividad según los aspectos o facetas del sistema (Giraldo, 2010)

Esta clasificación se justifica porque cada enfoque por separado no permite un adecuado modelado de todos los aspectos. UML (OMG, 1999), por ejemplo, se interesa por el modelado dentro de los límites de la aplicación; es por esto que provee el elemento de modelado en el diagrama de casos de uso denominado boundary (rectángulo que encierra los casos de uso del sistema). Para UML las interacciones entre los actores del sistema no son de interés en la especificación de requerimientos, lo cual es una gran limitación. Esta interacción puede ser modelada usando CIAN (Molina et al, 2008). (Inclusive las necesidades de comunicación, cooperación y colaboración entre roles).

3.2.2 Nivel de abstracción

La abstracción es un medio cognitivo por el que los ingenieros, matemáticos y otros se ocupan de la complejidad. Abarca tanto aspectos de la eliminación de detalles, así como la identificación de generalizaciones o características comunes, y ha sido reconocida como una habilidad crucial para los profesionales de la ingeniería del software (Kramer et al, 2006). Igualmente, se ha reconocido su importancia en el pensamiento computacional. Se hace énfasis en la necesidad de pensar en múltiples niveles de abstracción y en la capacidad de moverse entre dichos niveles (Wing, 2006). P. ej., los desarrolladores de software se mueven desde una vista global a una vista local, detallada del Sistema, y viceversa. Se reconoce también que entre estos dos niveles existen niveles de abstracción intermedios que los programadores consideran durante el proceso de desarrollo de software y que el conocimiento de cómo y cuándo moverse entre niveles de abstracción requiere algún grado de awareness y experiencia (KramerHazzan, 2006). Finalmente, lo que hace tan importante la práctica del pensamiento abstracto en ingeniería es que contribuye a desarrollar la capacidad de análisis y modelado formal. El modelado es la técnica más importante en ingeniería porque los modelos ayudan a comprender y analizar problemas complejos (Kramer, 2007).

De acuerdo con esto, se propone una división en dos categorías o niveles: nivel de negocio y nivel de sistema (Figura 3.2).




El nivel de negocio (Modelo de Negocio) se enfoca en capturar la esencia del sistema. Es un sistema libre de tecnología, en el que interesan las necesidades e intenciones de los usuarios y el propósito fundamental del negocio. El nivel de sistema informático (Modelo del Sistema) se centra en la definición de las actividades que están relacionadas con la interacción con el sistema, tanto desde la perspectiva del usuario como la del ordenador que las lleva a cabo.

Figura 3.2. Estructura de la clasificación de la actividad en un solo aspecto (Giraldo, 2010)

3.2.3 Nivel de granularidad

Cada nivel de abstracción puede ser clasificado en dos categorías, niveles de anidamiento o jerarquía entre las actividades: actividades de bajo nivel de granularidad y actividades del alto nivel de granularidad (Figura 3.2). Esta división obedece a la forma natural que tiene el ser humano de procesar el trabajo,



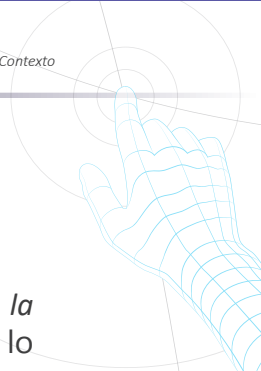
descomponiendo un problema en una serie de subproblemas cada vez más sencillos y con objetivos más fáciles de alcanzar. La descomposición de las actividades de baja granularidad tiene como finalidad la generación de un conjunto de actividades de alta granularidad que ya no serán más descompuestas, terminando el proceso de descomposición. Por ejemplo: una acción de negocio dentro de un diagrama de actividad puede transformarse en un diagrama de CTT (Paternò, 2004) (alta-baja granularidad) o puede transformarse en un caso de uso de sistema (alta-alta granularidad). Una vez se definen o identifican las tareas de alta granularidad en un nivel de abstracción en particular, se termina el diseño en dicho nivel y se pasa al nivel de abstracción siguiente. El principal reto con este clasificador está en la identificación de las actividades de alta granularidad. En la literatura no se han encontrado estudios que orienten en la identificación de actividades de alta granularidad. Sin embargo, Lauesen (Lauesen, 2005) provee una guía para la identificación de actividades buenas, malas y vagas, que puede ser adoptada para la definición de la actividad de alto nivel de granularidad en el nivel de negocio. Una guía para la definición de la actividad de alto nivel de granularidad en el nivel de sistema puede ser: las tareas de menor granularidad son aquellas tareas que cumplen con ciertos patrones y que tienen una complejidad relativa a las usadas para estudiar la asignatura de complejidad computacional, por ejemplo: crear, buscar, ordenar, abrir, borrar, etc.

3.2.4 Tipo de actividad

La clasificación por tipo de actividad intenta diferenciar la información que es esencial para el diseño de la interfaz de usuario y separarla de cualquier otra información considerada útil para otros fines. Las actividades se clasifican según su tipo en: Actividades de interacción con los usuarios (actividades base y de soporte) y Actividades sin interacción con el usuario (actividades de gestión) (Figura 3.2).

Las actividades base son funciones o procesos que tienen sentido para el usuario desde el punto de vista de sus intenciones o sus propósitos. P.ej., en un editor de texto, la función "editar" es una función base, mientras que la función "buscar" es una función de soporte. Ambos tipos de actividad deben tener los mismos requisitos de usabilidad y eficacia (IBM_Rational, 2003b). Las actividades de tipo soporte proporcionan funcionalidad básica para las actividades base, mientras que, las actividades de gestión son las responsables de la gestión de la empresa y del sistema. Por ejemplo, pueden servir para mantener la configuración, la navegación o el diálogo.

La Clasificación de la Actividad para un aspecto según el nivel de abstracción, la granularidad y el tipo, produce doce secciones con distintas descripciones de la actividad (Figura 3.2). Mediante una serie de flechas punteadas se indican, por un lado, las categorías relacionadas con el tipo de actividad y, por otro lado, la participación que existe entre algunos involucrados y las actividades. El actor de negocio interviene en las actividades del negocio que son de tipo base y de soporte; el trabajador de negocio interviene en todos los tipos de actividad del negocio; el actor de sistema interviene en las actividades del sistema que son de tipo base y de soporte y el control de sistema interviene en todos los tipos de actividad del sistema.



3.2.5 Razones para elegir la clasificación del modelado de la actividad

La palabra Taxonomía: *"Ciencia que trata de los principios, métodos y fines de la clasificación"*, está relacionada directamente con el significado de "Clasificación". Si lo que se pretende es definir una Taxonomía de la Actividad, se debe tener en cuenta entonces el concepto de la Clasificación de la Actividad o mejor Clasificación del Modelado de la Actividad como lo define (Giraldo, 2010), (Giraldo et al, 2014b).

El modelado de la actividad ha sido estudiado por varias disciplinas (la Psicología Cognitiva o la Ergonomía, la Planificación y Asignación de tareas, la Ingeniería del Software, Etnografía, Colaboración y otros (Limbourg, 2004)). Todas estas disciplinas estudian las actividades para asegurar la comprensión de cómo: los usuarios pueden interactuar con una interfaz de usuario para la realización de una tarea interactiva o como evaluar la carga de trabajo, para planificar y asignar tareas a los usuarios de una organización en particular, capturar información de las tareas en una forma operativa que sea comprensible por una máquina o cómo los seres humanos interactúan con una interfaz de usuario en un determinado contexto de uso.

Otra de las razones para tomar como base esta propuesta es que también encuentra sus bases en el estudio de múltiples reflexiones o propuestas que se originan en la teoría de la actividad (Kuutti, 1991) y que consideran a la actividad como concepto intermediario o mediador para comprender la relación entre conceptos de la teoría de la actividad como: individuo-división de trabajo, individuo-objeto, individuo-herramienta, etc. (Souza, 2003). Todas estas relaciones pueden definir un contexto distinto para representar la actividad, o viceversa, la actividad representa un contexto distinto para el estudio de estas relaciones.

3.3 FRAMEWORK DE ZACHMAN COMO BASE PARA LA DEFINICIÓN DE LA TXA

El Framework de Zachman (Sowa y Zachman, 1992) es uno de los framework más representativos a nivel de descripción de sistemas empresariales. Sin embargo en esta tesis se interpreta como una taxonomía que define un esquema que constituye el conjunto total de representaciones descriptivas relevantes para especificar cualquier cosa, cualquier sistema. Por esta razón, se toma como base su estructura taxonómica, sus reglas y principios.

Nos interesa la taxonomía de Zachman porque describe la Actividad "How" en distintos niveles de abstracción y cómo se dan las transformaciones entre dichos niveles (verticalmente), en lugar de cambiar el nivel de especificación. Utiliza el concepto de Integración a nivel horizontal y esta es una de las principales características que debe tener la Taxonomía de la Actividad que se propone en esta tesis. Se pretende que dicha taxonomía sea una propuesta integradora, que no sea una propuesta aislada, sino que esté relacionada con los demás modelos que representan un sistema interactivo

completo y que sean analizados, implementados según un interés específico. No es suficiente un Modelado de la Actividad si no está atado a los modelos de la Organización, de los Datos, de los Lugares, etc.

3.4 MAPA DE RUTA PARA LA DEFINICIÓN DE LA TXA

Los fundamentos teóricos y conceptuales, además de las propuestas que conforman los antecedentes para esta tesis, se toman como base a partir de la cual se define el mapa de ruta para la definición de la Taxonomía de la Actividad (Figura 3.3).

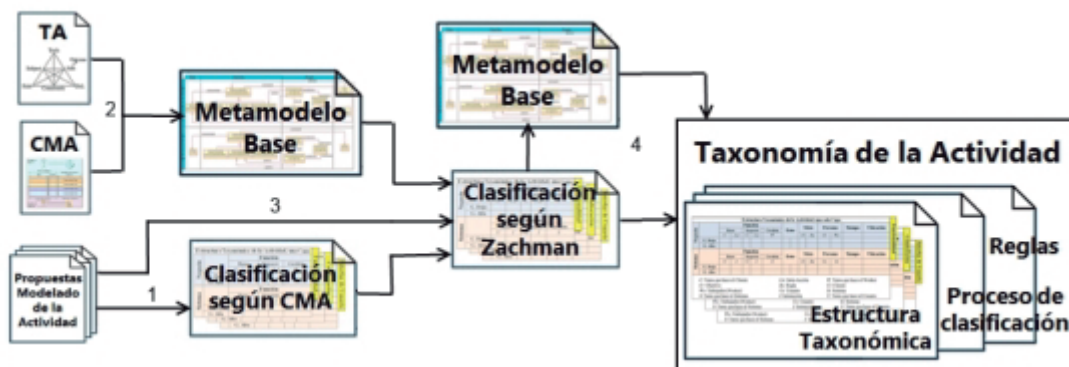
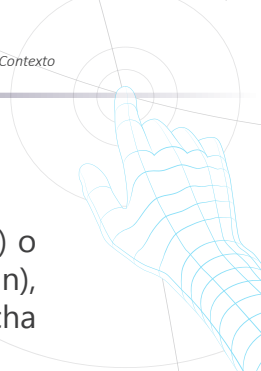


Figura 3.3. Mapa de ruta para la definición de la TxA

El mapa de ruta que se expone en la Figura 3.3 sigue la siguiente secuencia: Primero, se realiza un análisis de las propuestas que se enfocan en el modelado de la actividad para sistemas interactivos, de acuerdo a la taxonomía propuesta por la Clasificación de la Actividad. Segundo, se define un metamodelo base a partir de la Teoría de la Actividad y de la Clasificación de la Actividad. Es importante resaltar que para la definición de este metamodelo se sigue la propuesta para la integración de procesos y notaciones definida por Giraldo (Giraldo, 2010). Tercero, se realiza un análisis de las propuestas que se enfocan en el modelado de la actividad para sistemas interactivos, de acuerdo a la Taxonomía de Zachman, con el fin de enriquecer el metamodelo base y los clasificadores obtenidos en el segundo paso. Por último, se definen los clasificadores complementarios que resultan de un análisis más profundo de todas las propuestas.

3.4.1 Clasificación de propuestas mediante el uso de la clasificación de la actividad

Como se mencionó anteriormente, el mapa de ruta sigue una secuencia. El primer paso consiste en la clasificación de las propuestas que se enfocan en el modelado de la actividad para sistemas interactivos, de acuerdo a la taxonomía propuesta por la Clasificación de la Actividad. La clasificación se enfoca en los elementos de modelado que conforman cada una de las propuestas estudiadas. La descripción realizada en la



sección 2.2 es la base para catalogar cada propuesta de acuerdo al/ los aspecto (s) o faceta (s) en el/ los que se enfoca para especificar la actividad, Funcionalidad (Fun), Colaboración (Collab) Interfaz de Usuario (UI). La Tabla 3.1 muestra en resumen dicha clasificación.

Tabla 3.1. Aspectos en los que se enfocan las propuestas analizadas

Proposal/Aspect	Fun	Collab	UI
(1) Zachman Framework (Sowa et al, 1992).	X		
(2) BPMN (OMG, 2011)	X		
(3) UML-RUP-(OMG, 2008) OpenUp (Balduino, 2007)	X		
(4) CIAN-CIAM (Molina et al, 2008)		X	
(5) CTT (Paternò, 2004)			X
(6) HAMSTERS (Barboni et al, 2010)		X	X
(7) OASIS (Letelier et al, 1998)	X		
(8) OO Method (Pastor et al, 2001)	X		
(9) Amenities (Garrido, 2003)		X	
(10) Wisdom (Nunes, 2001)	X		X
(11) Model Driven Software Development Framework (Cubillo, 2011)	X		
(12) UCD with Activity Modeling (Constantine, 2006)			X
(13) TOUCHE (Penichet, 2007)		X	X
(14) Traetteberg (Trætteberg, 2002)			X
(15) Methodology for Developing UI to Workflow IS (Garcia, 2010)			X
(16) Limbourg (Limbourg, 2004)			X
(17) Dygimes (Luyten, 2004)			X
(18) IDEAS (Lozano, 2001)			X
(19) MPIu+a (Granollers, 2004)			X
(20) CIAF (Giraldo, 2010)	X		X

La clasificación de elementos de modelado de cada propuesta de acuerdo a la taxonomía, tiene como fin evaluar la capacidad de cada propuesta metodológica para representar las actividades que realizan los sistemas modelados por las mismas (Tablas 3.2 a 3.4). Se clasifica la notación y no la metodología porque la notación es la que describe la información del producto, en este caso el sistema interactivo.

La información plasmada en las Tablas 3.2 a 3.4 permite identificar los elementos de modelado que cada propuesta proporciona para especificar un sistema interactivo de acuerdo con las dimensiones que propone la Taxonomía de Giraldo, Aspecto, Nivel de Abstracción, Nivel de Granularidad y Tipo de Actividad.

Se observa que a diferencia de las demás propuestas, el **OpenUp**, **BPMN**, **Model Driven Software Development Framework** y **TD-MBUID** proveen elementos de modelado de la actividad en los que se separa claramente los niveles de abstracción

Negocio y Sistema. Aunque algunas de las demás propuestas proveen elementos de modelado que se ubican en los dos niveles de abstracción, dichas propuestas no lo especifican explícitamente.

Tabla 3.2. Clasificación taxonómica de la actividad, capa funcionalidad

Activity Taxonomic Classification, Functionality Layer				
		Function		
		Base	Support	Management
		<i>Client Task</i>	<i>Client Task</i>	<i>Worker Task</i>
Business	Low Granularity	(1)<Business Process>. (2)<Abstract Task>, <Manual Task>. (3):<Business UC>, <Business Activity>. (10):<Essential UC>, <Association>, <Activity>. (11):<Communicative Event>, <Specialized Communicative Event>, <Precedence Relation>. (20):<Business UC>, <Business Activity>.	(1) <Business Process>. (2)<Abstract Task>. (3):<Business UC>, <Business Activity>. (10):<Essential UC>, <Association>, <Activity>. (11):<Communicative Event>, <Specialized Communicative Event>, <Precedence Relation>. (20):<Business UC>, <Business Activity>.	(1) <Business Process>. (2)<Abstract Task>, <Service Task>, <Script Task>. (3):<Business UC>, <Business Activity>. (10):<Essential UC>, <Association>, <Activity>. (11):<Communicative Event>, <Specialized Communicative Event>, <Precedence Relation>. (20):<Business UC>, <Business Activity>.
	High G.	(1) <Business Function>. (2) <Subprocess>, <Transaction>. (3):<Business Action>. (20):<Business Action>.	(1) <Business Function>. (2) <Subprocess>, <Transaction>. (3):<Business Action>. (20):<Business Action>.	(1) <Business Function>. (2) <Subprocess>, <Call Activity>. (3):<Business Action>. (10):<Control>. (20):<Business Action>.
System		Function		
		Base	Support	Management
		<i>User Task</i>	<i>User Task</i>	<i>System Task</i>
System	Low G.	(1)<System Process>, (2)<User Task>, <Subprocess>. (3):<Compound System UC>. (20):<Compound System UC>.	(1)<System Process>. (2)<User Task>, <Subprocess>. (3):<Compound System UC>. (20):<Compound System UC>.	(1)<System Process>, <Subprocess>. (3):<Compound System UC>. (8):<Service> (20):<Compound System UC>.
	High G.	(1)<Information System> (2)<Transaction>. (3):<SystemAction>. (20):<SystemAction>.	(1)<Information System>. (2)<Transaction>. (3):<SystemAction>. (20):<SystemAction>.	(1)<Information System> (2)<Call Activity>. (3):<SystemAction>, <Control>. (7),(8):<State>,<Operation>, <Trigger> (10):<Boundary>. (20):<SystemAction>, <Control>.

En general, la mayoría de propuestas utilizan la notación **CTT** (Paternò, 2004) para realizar modelado de tareas y la notación **UML** (OMG, 1999) para realizar modelado de la funcionalidad y de dominio. Dichas notaciones se utilizan también como base cuando se requiere introducir nuevos elementos de modelado como se hace por ejemplo en **WISDOM**, **IDEAS** y **HAMSTERS**.



La distinción entre tipos de actividad sólo es realizada por las propuestas que se basan en la notación **CTT**. Se hace la distinción entre tarea de usuario (Base y Soporte) y tarea de sistema (Gestión), aunque sólo en el nivel de abstracción de Sistema. Adicionalmente, no se realiza la distinción sintáctica para soportar de manera separada la clasificación por tipo de actividad en las categorías Base y Soporte. Es decir, para cualquier tipo de actividad se usarían los mismos elementos de modelado. Y aunque es posible clasificar las actividades de un sistema informático en dichas categorías, esta falta de elementos de modelado para cada propuesta genera un bajo nivel de expresividad y semántica para modelar cualquier aspecto o faceta distintos a los que modela cada propuesta.

Tabla 3.3. Clasificación taxonómica de la actividad, capa colaboración

Activity Taxonomic Classification, Collaboration Layer				
		Function		
		Base	Support	Management
		<i>Client Task</i>	<i>Client Task</i>	<i>Worker Task</i>
Business	Low G.	(4):<Inter-action Task>. (20):<Inter-action Task>, <Cooperative Task>, <Collaborative Task>.	(4):<Inter-action Task>. (20):<Inter-action Task>, <Cooperative Task>, <Collaborative Task>.	(4):<Inter-action Task>. (20):<Inter-action Task>
	High G.	(20):<Inter-action Task>, <Cooperative Task>, <Collaborative Task>.	(20):<Inter-action Task>, <Cooperative Task>, <Collaborative Task>.	
System		Function		
		Base	Support	Management
		<i>User Task</i>	<i>User Task</i>	<i>System Task</i>
	Low Granularity	(4):<Cooperative Task>, <Collaborative Task>. (6):<User Cooperative Abstract Task>, <Abstract Group Task>. (9):<Task>,<Subactivity>, <Rol>,<Organization> (20):<Cooperative Task>, <Collaborative Task>.	(4):<Cooperative Task>, <Collaborative Task>. (6):<User Cooperative Abstract Task>, <Abstract Group Task>. (9):<Task>,<Subactivity>,<Rol>, <Organization> (20):<Cooperative Task>, <Collaborative Task>.	(9):<Rol>,<Organization>
High Granularity	(6):<User Cooperative Processing>, <User Individual Group Task>, <User Cooperative Group Task>. (9):<Action>	(6):<User Cooperative Processing>, <User Individual Group Task>, <User Cooperative Group Task>. (9):<Action>	(20):<System Group Task>. (9):<Action>	

Es interesante cómo en la propuesta metodológica (Guerrero, 2010) se introducen elementos de **modelado de Workflow** relacionados con la ubicación (dónde se ejecuta la actividad), como "*Organizational Unit*" y también elementos relacionados con el tiempo (cuándo se ejecuta la actividad), como el concepto de "*Agenda*". Elementos que podrían integrarse a la definición de la TxA para tener una especificación más

completa de la Actividad en los sistemas interactivos.

Es importante resaltar también que propuestas como las de **TOUCHE** e **IDEAS** proveen un conjunto de plantillas para capturar información relacionada con la actividad en los sistemas interactivos. Y aunque no se provee alguna notación en dicha especificación, es útil analizar cómo se podría integrar esta información a la Taxonomía de la Actividad propuesta.

Tabla 3.4. Clasificación taxonómica de la actividad, capa interfaz de usuario

Activity Taxonomic Classification, User Interface Layer				
		Function		
		Base	Support	Management
		<i>Client Task</i>	<i>Client Task</i>	<i>Worker Task</i>
Business	Low G.	(14):<Supertask> (15):<Place>,<Transition>,<Arc>. (19):<Task>,<Storyboard>. (20),(5):<CIAN Abstract Task>, <CTT Abstract Task>, <Individual Task>	(14):<Supertask>. (15):<Place>,<Transition>,<Arc>. (19):<Task>,<Storyboard>. (20),(5):<CIAN Abstract Task>, <CTT Abstract Task>, <Individual Task>	(14):<Supertask>. (15):<Place>,<Transition>,<Arc>. (19):<Task>,<Storyboard>.
	H.G.	(14):<Sub-task>. (19):<Sub-Task>,<Storyboard>, <Navigational Storyboard>.	(14):<Sub-task>. (19):<Sub-Task>,<Storyboard>, <Navigational Storyboard>.	(14):<Sub-task>. (19):<SubTask><Storyboard>< Navigational Storyboard>.
System	Low Granularity	Base	Support	Management
		<i>User Task</i>	<i>User Task</i>	<i>System Task</i>
		(12):<Activity>,<Relationship>. (4),(5):<CIAN Abstract Task>, <CTT Abstract Task>. (13),(5):<System Use Case>, <CTT Abstract Task>. (15),(5):<CTT Abstract Task>. (18):<Use Case>, <Dialog Interaction Diagram>. (6):<Abstract Task>, <User Individual Abstract Task>. (19),(5):<CTT Abstract Task>, <Scenes>,<Use Cases>. (20),(5):<CIAN Abstract Task>, <CTT Abstract Task>.	(12):<Activity>,<Relationship>. (4):<CIAN Abstract Task>, <CTT Abstract Task>. (13):<System Use Case>, <CTT Abstract Task>. (15):<CTT Abstract Task>. (18):<Use Case>, <Dialog Interaction Diagram>. (6):<Abstract Task>, <User Individual Abstract Task>. (19),(5):<CTT Abstract Task>, <Scenes>,<Use Cases>. (20):<CIAN Abstract Task>, <CTT Abstract Task>.	(10):<Task>. (4),(5):<CTT Abstract Task>. (13),(5):<System Use Case>, <CTT Abstract Task>. (15),(5):<CTT Abstract Task>. (18):<Use Case>, <Dialog Interaction Diagram>. (6):<System Abstract Task>. (19):<CTT Abstract Task>, <Scenes>,<Use Cases>. (20),(5):<CTT Abstract Task>.
High Granularity	(12):<Action> (4),(5):<Tool>, <CTT User Task>. (13),(19),(5):<CTT User Task>. (15),(5):<CTT User Task>, <CTT Cooperative Task>. (16),(5):<CTT User Task>. (17),(5):<CTT User Task>, <State>,<Transition>,<Event>. (18):<Sequence Diagram>, <Internal States Diagram>. (6):<User Individual Processing>. (20),(5):<Tool>, <CTT User Task>.	(12):<Action> (4),(5):<Tool>, <CTT User Task>. (13),(19),(5):<CTT User Task>. (15),(5):<CTT User Task>, <CTT Cooperative Task>. (16),(5):<CTT User Task>. (17),(5):<CTT User Task>, <State>,<Transition>,<Event>. (18):<Sequence Diagram>, <Internal States Diagram>. (6):<User Individual Processing>. (20),(5):<Tool>, <CTT User Task>.	(4),(5):<CTT Application Task>. (13),(19),(5):<CTT Application Task>. (15),(5):<CTT Application Task>. (16),(5):<CTT Application Task>. (17),(5):<CTT Application Task>, <State>,<Transition>,<Event> (18):<Sequence Diagram>, <Internal States Diagram>. (6):<System Processing>. (20),(5):<CTT Application Task>.	

Se observa también, que las propuestas analizadas proveen pocos elementos de modelado de la actividad en el nivel de abstracción de negocio, tanto en el nivel alto como bajo de granularidad. Es necesario analizar entonces si estos elementos serían suficientes para modelar la actividad en este nivel y también qué otros elementos de modelado se podrían definir para diferenciar los tipos de actividad. En el nivel de abstracción del sistema se observa una cantidad mayor de elementos de modelado que los que aparecen en el nivel de abstracción de negocio, lo que significa que el conjunto de las propuestas se centra más en detallar la información de los sistemas interactivos en este nivel. Sin embargo, no sólo la cantidad de elementos de modelado debe ser el único indicador válido, por lo tanto, es necesario analizar otros elementos o aspectos o atributos de la clasificación, por ejemplo, desde el punto de vista de la calidad de las notaciones podría ser de interés analizar el nivel de expresividad, la economía, la completitud, redundancia, etc (Moody, 2010), además de las capacidades que tiene cada lenguaje para la integración.

Finalmente, la recopilación de los elementos de modelado clasificados, permite identificar qué conceptos serían necesarios para incorporarlos en la taxonomía, es decir, para enriquecerla por ejemplo, con otros clasificadores y que permitan especificar con más precisión los sistemas interactivos que se pretendan modelar. La clasificación realizada también permite identificar puntos de integración entre las propuestas, como se planteó en la **Estrategia 3**, sección 1.2.

3.4.2 Integración de propuestas para la definición del metamodelo base

Una vez identificada la información y los clasificadores que van a hacer parte de la taxonomía, es necesario definir un metamodelo base que será tenido en cuenta para la estandarización de la información en cada vista de la estructura taxonómica. Esto corresponde con el paso 2 del mapa de ruta (Figura 3.3).

La definición del metamodelo base está soportada por el proceso de integración de procesos y notaciones definido por Giraldo (Giraldo, 2010). Dicho proceso tiene un buen nivel de formalidad y es organizado para clasificar e integrar los elementos conceptuales de las propuestas analizadas. El proceso de integración comprende 5 pasos, a saber: 1. Identificación de elementos conceptuales de interés; 2. Normalización de los elementos conceptuales; 3. Identificación de puntos de intercambio; 4. Definición de clasificadores; y 5. Definición del metamodelo base. A continuación se describe en detalle el proceso aplicado a la definición de nuestro metamodelo base.

3.4.2.1 Identificación de elementos conceptuales de interés

Las propuestas que se someten a este método de integración, son la Teoría de la Actividad (Engeström, 2001) y la Clasificación del Modelado de la Actividad (Giraldo, 2010). Ambas propuestas se fundamentan en la especificación de la Actividad en diferentes áreas del conocimiento.

Los elementos conceptuales que define la Teoría de la Actividad son: División de la labor, Propósito, Sujeto, Herramientas, Comunidad, Reglas y Resultado.



Por su parte, la Clasificación de la actividad define los elementos conceptuales: Actividad de tipo base, Actividad de tipo soporte, Actividad de tipo gestión, Actor de negocio, Trabajador de negocio, Actor de sistema, Sistema, Nivel de abstracción de negocio, Nivel de Abstracción de sistema, Granularidad alta, Granularidad Baja,

3.4.2.2 Normalización de elementos conceptuales empleando un mismo estándar

Una vez identificados los conceptos, se requiere identificar cómo están relacionados. La estructura de un Sistema de Actividad se encuentra expresada con una figura en forma de triángulo (Figura 2.1). Adicionalmente, los conceptos que definen la Teoría de la Actividad se describen en lenguaje natural. Existen entonces posibles ambigüedades tanto en la interpretación de los conceptos como en su aplicación a una situación en particular. Nuestro propósito es que la TA se adapte al desarrollo de software para que soporte en alto grado la automatización de la labor. Por lo tanto, se requieren definiciones e interpretaciones precisas de los conceptos relacionados con la Taxonomía de la Actividad. Por otro lado, la estructura de la Clasificación de la Actividad se plasma en una matriz de dos dimensiones (Figura 3.2). Es claro entonces que el nivel de representación de ambas propuestas difiere en gran medida, lo cual hace difícil identificar relaciones de mapping y trazabilidad entre los conceptos que especifican. Se considera necesario utilizar un mismo nivel de representación de conceptos. Para esto, se utiliza el estándar de metamodelado expresado en el lenguaje UML porque permite tener una vista clara de los conceptos y sus relaciones facilitando la identificación de relaciones de mapping entre las propuestas analizadas. De esta forma se llevan los elementos conceptuales a un mismo nivel de representación.

El metamodelo que representa la estructura de un Sistema de Actividad es tomado de la propuesta definida por Georg (Georg et al, 2015). Este metamodelo contiene los elementos conceptuales definidos por la TA extendida y define las relaciones y las restricciones que existen entre ellos (Figura 3.4).

En la figura 3.4 se observa que los elementos conceptuales y sus relaciones, representan la estructura de un Sistema de Actividad. Sin embargo el elemento conceptual "*Actividad*" no se considera dentro de dicha estructura. Esto indica que existe una dicotomía en el sentido de que se toma la "*Actividad*" como algo que tiene entidad, que es el "*Todo*" pero que para que sea entendible, necesita ser dividido en dos partes complementarias: "*Actividad*" y sus "*Elementos Estructurales*".

Según el análisis realizado a la propuesta "*Clasificación del Modelado de la Actividad*", se propone un metamodelo que representa el marco conceptual de la Clasificación del Modelado de la Actividad (Figura 3.5). Nótese que los elementos conceptuales pertenecen a un solo metamodelo, aunque se deja expresado de manera separada en los niveles de abstracción Negocio y Sistema. Se entiende que la información que se captura y se describe, a través del metamodelo, no debe mezclar ni combinar niveles de abstracción. La información contenida en el cada nivel de abstracción finalmente representa lo mismo, pero la relación existente entre dichos niveles debe estar expresada en términos de mapping.

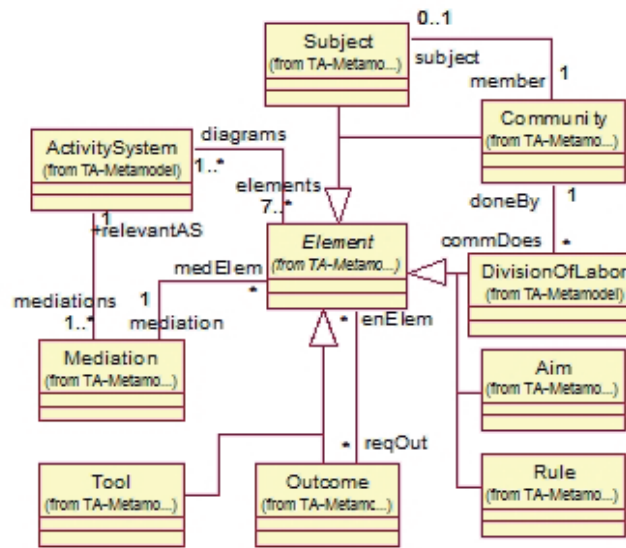


Figura 3.4. Metamodelo de la TA definido en (Georg, 2015).

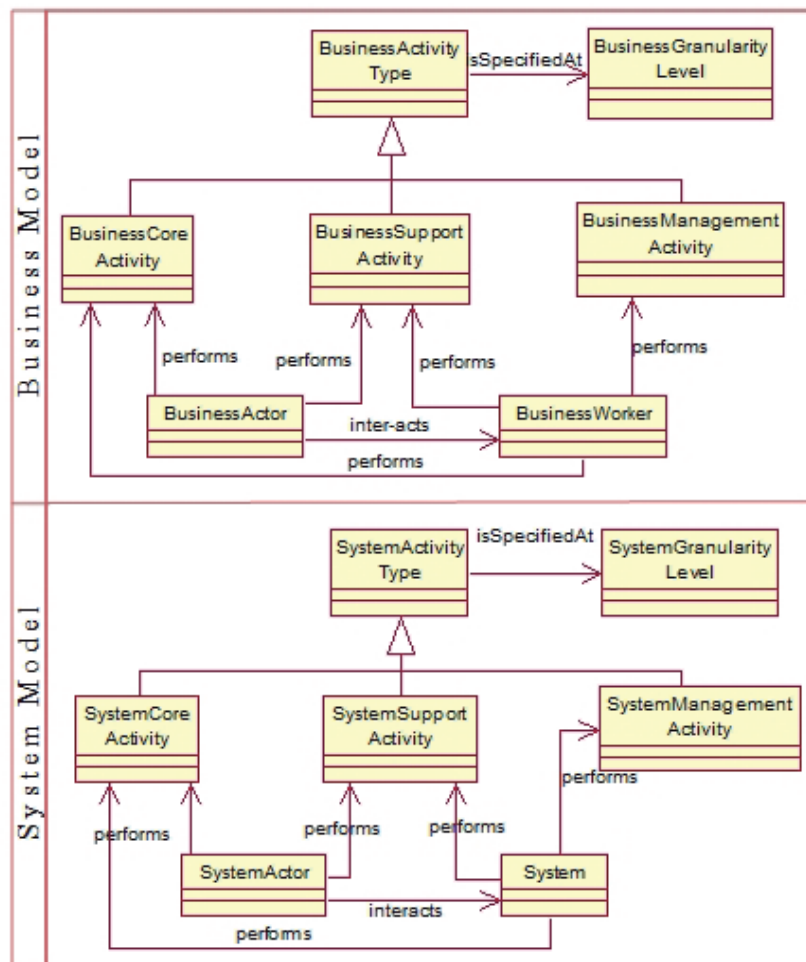


Figura 3.5. Metamodelo de la Clasificación de la Actividad

3.4.2.3 Identificación de puntos de intercambio

Un punto de intercambio es una o más piezas de información de un marco conceptual que poseen cierta correspondencia estructural o sintáctica con otras piezas de información en otro marco conceptual (Giraldo, 2010). De acuerdo con esta definición, el propósito en esta etapa es identificar tanto los elementos conceptuales de la Clasificación del Modelado de la Actividad, como los elementos conceptuales de la Teoría de la Actividad que posean una correspondencia entre sí. Mediante este enfoque de integración se pretende encontrar el mapping semántico entre las propuestas y posteriormente identificar la correlación existente entre los elementos conceptuales. Es necesario analizar los mappings existentes entre las propuestas base para establecer cómo se relacionan y para definir posibles elementos faltantes que contribuyan a completar la TxA.

Los principales elementos conceptuales de la Clasificación del Modelado de la Actividad que tienen mapping con la TA son los que se refieren a los Tipos de Actividad y los Roles que las realizan tanto a nivel de Negocio como a nivel de Sistema (Figura 3.6). El mapping entre estas dos propuestas muestra que desde la perspectiva del Negocio, la TA se centra en los clasificadores Persona (*Community, Subject*) y Actividad (*DivisionOfLabor*); y que para el nivel de abstracción Sistema, la TA se centra en Instrumento (*Tool*). La Clasificación del Modelado de la Actividad, por su parte, se centra en separar los niveles de abstracción Negocio y Sistema para todos sus clasificadores, Persona (*Actor, Worker, Sistema*) y Actividad (*Core, Support, Management*). La actividad a su vez se especifica en niveles de granularidad.

El metamodelo de la TA expresa que cada instancia de Sujeto (*Subject*) está relacionada con una instancia de Comunidad (*Community*). Nótese que no se tiene alguna relación de mapping desde el metamodelo de la Clasificación del Modelado de la Actividad hacia la metaclassa Sujeto (*Subject*) del metamodelo de la TA. Lo que se hace es un mapping desde el elemento conceptual Comunidad, hacia los elementos conceptuales Actor de Negocio (*BusinessActor*) y Trabajador de Negocio (*BusinessWorker*).

En esta tesis se pretende utilizar la TA y la Clasificación del Modelado de la Actividad como base para la definición de un marco conceptual. Este marco conceptual define la sintaxis abstracta a partir de la cual se define la sintaxis concreta para la creación de notaciones enfocadas en el modelado de la Actividad. De acuerdo con este propósito, Giraldo (Giraldo, 2010) hace mención a que una condición deseable en un marco conceptual es que se proporcione un conjunto de conceptos asociados a un mismo clasificador, para así representar modelos sobre una vista y una perspectiva en particular.

El análisis realizado a los metamodelos de la TA y la Clasificación del Modelado de la Actividad (Figura 3.6), permite observar que para el caso de la TA, el metamodelo sólo proporciona más de un elemento conceptual para el clasificador Persona (*Community* y *Subject*). Es decir, sólo sería posible especificar una sintaxis concreta (diagrama), para representar la relaciones existentes entre Comunidad y Sujeto. Sería imposible establecer un diagrama que represente de manera jerárquica las tareas que tienen

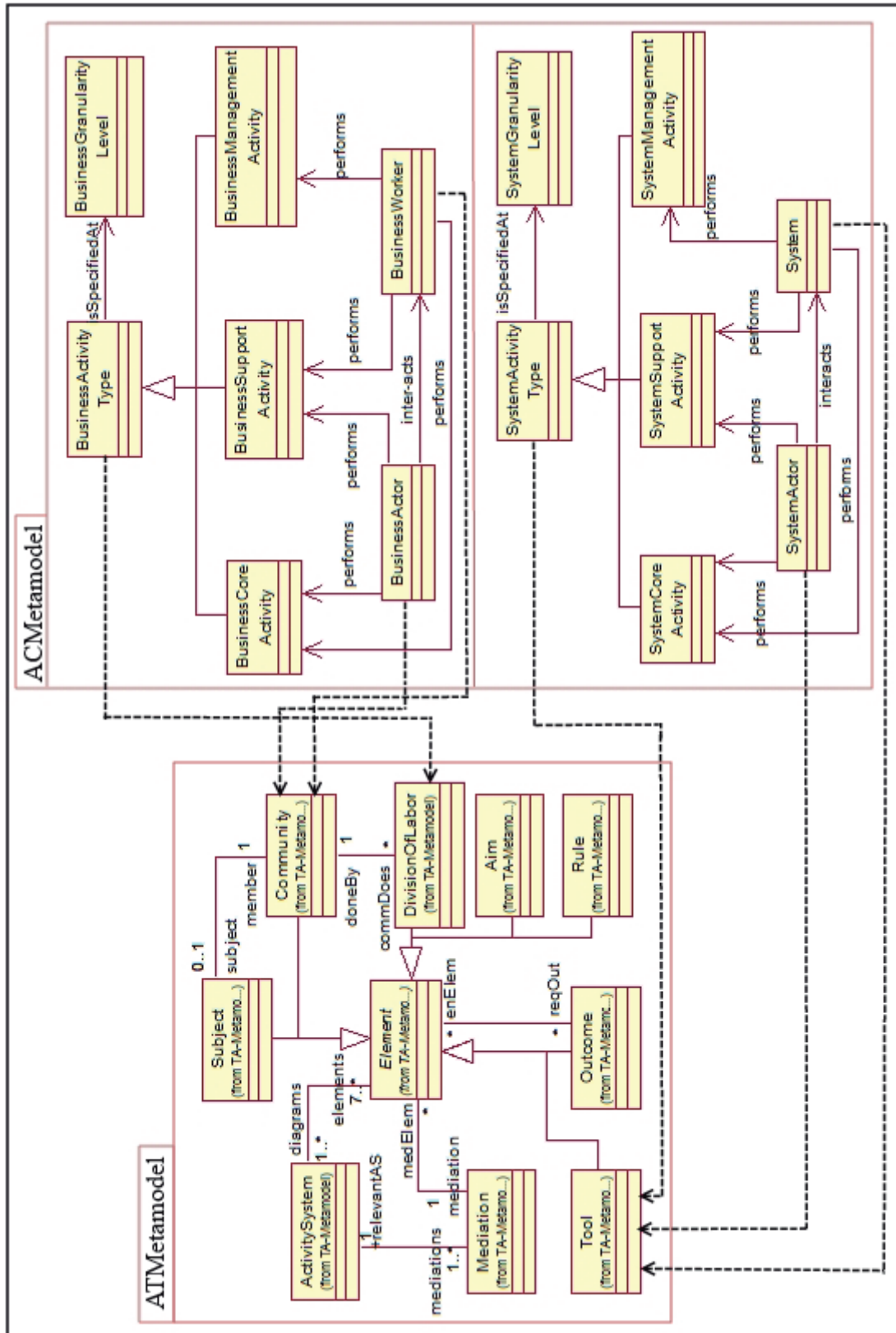


Figura 3.6. Mappings entre la TA y la Clasificación de la Actividad



en una comunidad. Para esto, sería necesario que la TA incluyese conceptos como el de tarea y relación entre tareas, como sí lo hace la Clasificación del Modelado de la Actividad. Adicionalmente, en el metamodelo de la TA (Figura 3.4) se observa que el único elemento conceptual utilizado para representar información asociada al clasificador Actividad, es División de la Labor (*DivisionOfLabor*). En esta misma línea, el elemento conceptual Herramienta (*Tool*) representa toda la información que se ubica en el nivel de abstracción Sistema. De esta forma, tanto "*División de la Labor*" como "*Herramienta*", sólo serían útiles para describir el nombre y algunos atributos del elemento de modelado que se desee representar a partir de cada uno de estos elementos conceptuales. En la mayoría de los casos esto es insuficiente. Por ejemplo, de la Ingeniería de Software se sabe que son muchos los conceptos que son indispensables para describir un sistema informático, por lo que, sólo a partir del elemento conceptual Herramienta (*Tool*) sería imposible representar los datos, los requisitos, la distribución, los roles, etc. necesarios para realizar dicha descripción.

3.4.2.4 Definición de los Clasificadores o Conceptos de cada celda

En esta sección, es necesario definir una capa de integración entre propuestas, al igual que se hace en el proceso de integración que se está siguiendo, propuesto por (Giraldo, 2010).

Según la información obtenida en el mapeo entre la TA y la Clasificación de la actividad (CA), la capa de integración contiene la siguiente información: perspectivas de modelo de negocio y modelo de sistema y las vistas de datos, función, personas y motivación. La intersección de vistas y perspectivas da lugar a 8 celdas de modelado (Figura 3.7).

Los conceptos de cada celda tienen la propiedad de ser una generalización de los elementos de modelado que representan con el fin de proporcionar un soporte adecuado a la información de dichos elementos de modelado para las distintas notaciones (Giraldo, 2010).

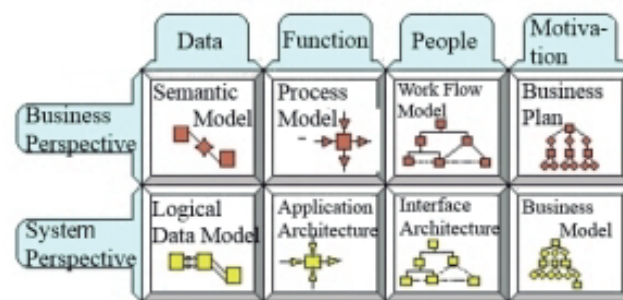


Figura 3.7. Capa de Integración para Análisis de Clasificadores

En la Figura 3.5 se observa la definición del concepto "Función" (*Function*) que generaliza los elementos de modelado "DoL" y "Tipo de Actividad de Negocio" (*BusinessActivityType*), a nivel de Negocio y los elementos de modelado "Herramienta" (*Tool*) y "Tipo de Actividad de Sistema" (*SystemActivityType*), a nivel de Sistema. Se observa también la definición del concepto "Personas" (*People*) que generaliza los elementos de modelado "Comunidad" (*Community*), "Sujeto" (*Subject*), "Actor de Negocio" (*BusinessActor*) y "Trabajador de Negocio" (*BusinessWorker*), a nivel de Negocio y los elementos de modelado "Herramienta" (*Tool*), "Actor de Sistema"

(SystemActor), "Sistema" (System), a nivel de Sistema. La relación denominada "suPersona" (itsPeople) generaliza el elemento de modelado "Mediación" (Mediation), el cual representa una relación de interconexión entre vistas.

El concepto "Persona" (People) está en capacidad de almacenar información tanto de un Sujeto, de una Comunidad, de un Trabajador de Netocio, como de un Actor de Negocio que esté disponible para efectos de la integración. La relación de mapeo indica que se debe especificar un concepto que sea capaz de generalizar todos los conceptos que mapea.

Es importante resaltar que la vista de la Función es la vista donde se deben ubicar los clasificadores de la Actividad "Cómo" (How), tema principal de interés para este trabajo. Las relaciones entre conceptos (Figura 3.8), permite observar que si nos referimos a la estructura conceptual de la Teoría de la Actividad (TA), en la Vista de la Función, esta teoría sólo define dos clasificadores que se ubican en esta vista: *División de la Labor* a nivel de Negocio y Herramienta (Tool) a nivel de Sistema. Los demás clasificadores definidos por la TA se ubican en otras vistas distintas a la de Función. Desde esta perspectiva, la TA se toma como un elemento integrador (Amalgama) entre elementos conceptuales de las demás vistas, en el proceso de definición de la Taxonomía de la Actividad.

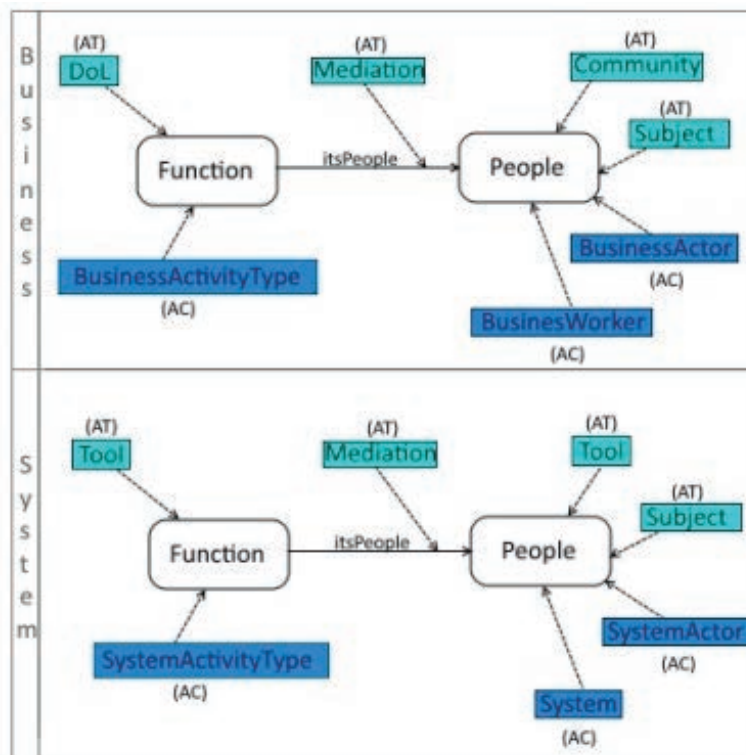


Figura 3.8. Relación de Conceptos entre la TA, la CA y la Capa de Integración

El análisis completo, tanto de las relaciones entre conceptos como de los mapeos, permite generar los clasificadores de cada celda de la capa de integración (Figura 3.9).

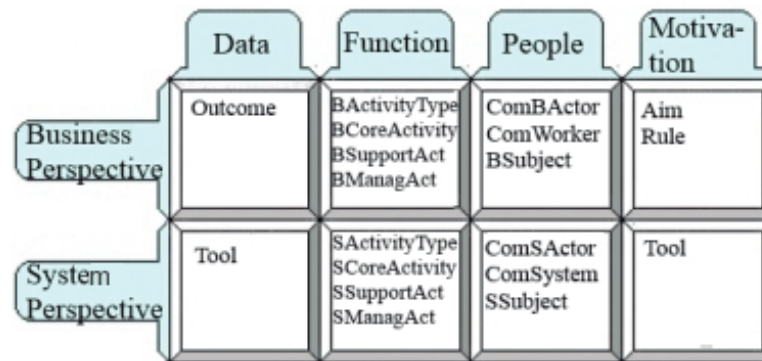


Figura 3.9. Clasificadores para Cada Celda de la Capa de Integración

El interés de definir estos clasificadores es que cada celda provea un contenedor para los modelos que abordan una determinada perspectiva y vista. De esta forma será posible proveer una representación desde distintos puntos de vista, en diferentes niveles de granularidad, generalidad y abstracción que posea lo necesario para el modelado de roles, interfaces, colaboración, tareas, y datos.

3.4.2.5 Definición del Metamodelo Base

El metamodelo base se especifica con el fin de mostrar las relaciones existentes entre los clasificadores ubicados en cada celda de la capa de integración (Figura 3.10). A partir del metamodelo se definirá la Estructura Taxonómica para la Taxonomía de la Actividad para Sistemas Interactivos.

Como lo define (Giraldo, 2010), cada concepto del metamodelo ubicado en una celda de la capa de integración debe ser completamente independiente de los conceptos de las otras celdas, es decir, cada elemento conceptual debe tener *independencia taxonómica*. Esto hace posible el hecho de generar notaciones con las que se especifiquen modelos de objetivos, de organización, de actividad, de datos y definir las relaciones estructurales entre dichos modelos.

3.4.3 Clasificación de propuestas según la taxonomía de Zachman

El análisis de la Teoría de la Actividad extendida (Engeström, 2001), como fuente de información primaria, junto con la Clasificación de la Actividad (Giraldo, 2010) permitieron la definición de un conjunto de clasificadores, y a partir de este conjunto, un Metamodelo Base para la definición de la Taxonomía de la Actividad. Dicho metamodelo contiene elementos de información clasificados en las celdas de "Datos", "Función", "Personas" y "Motivación". Aunque según como lo define el Framework de Zachman (Zachman, 1987), deberían tenerse en cuenta clasificadores para las celdas "Ubicación" y "Tiempo".

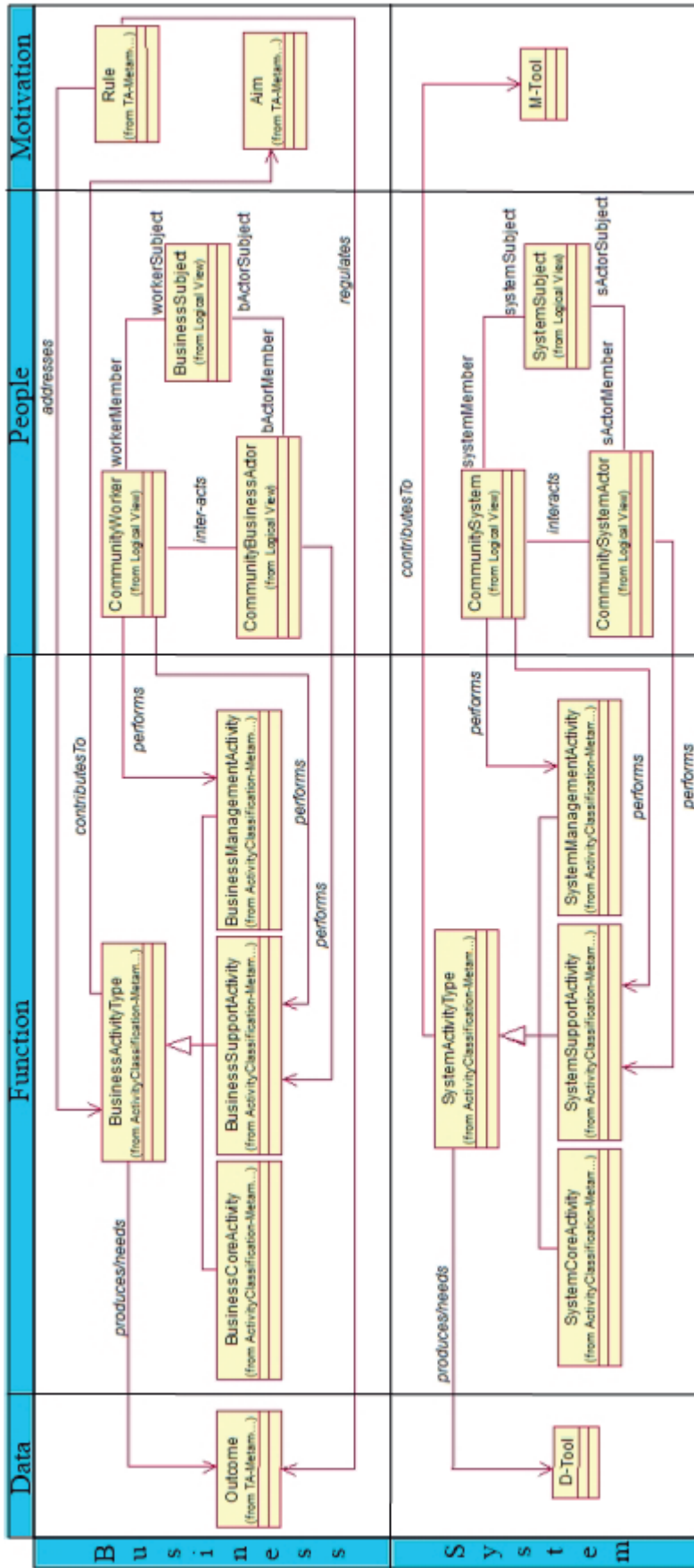


Figura 3.10. Metamodelo Base para definir la Estructura Taxonomica para la TxA



La Clasificación de propuestas mediante el uso de la Taxonomía de la Actividad Inicial, presentada en la sección 3.4.1, se enfoca únicamente en la celda "Función". A continuación se presenta la clasificación complementaria de los elementos de modelado del conjunto de propuestas estudiadas. Se tienen en cuenta las demás celdas que son necesarias para especificar la actividad (Tablas 3.5 a 3.7).

Tabla 3.5. Clasificación Taxonómica de la Actividad para Datos, Motivación, Personas, Tiempo y Ubicación, Capa Funcionalidad

Activity Taxonomic Classification, Functionality Layer						
	Data	Goal	Who	When	Where	
Business	Low G.	(1)<Business Entity>, <B.Subject Area>, <B. Relationship> (2)<Data Object>, <Data Input>, <Data Output>. (3):<Business Entity>. (10):<Entity>, <Communicate>. (11):<Communicative Interaction>. (20):<B. Entity>.	(1)<B.Plan>,<B.Rule> (3),(20):<B. Goal>. (2)<B. Rule Task>, <Gateway>.	(1)<B. Work Unit>. (2)<Pool>. (3):<B.Actor>, <B.Worker>. (10)<HumanActor>, <Association>. (11):<Primary/Receiver Actor>. (20):<B. Actor>.	(1)<B.Cycle> (2)<Sequence Flow>.	(1)<Business Location>. (3)<Node>.
	High G.	(1)<B. Property>, <B. Identifier>, <B. Fact>. (2)<Data Object>, <Data Input>, <Data Output>	(1)<B. Objective>, <B.Product/Service>, <B.Resource>, <B.Policy>, <External Regulation>. (2)<Gateway>	(1)<Job Position> (2)<Lane>. (3):<Swimlane>. (20):<Swimlane>.	(1)<B.Event>(2) <Start Event>, <Intermediate Event>, <End Event>.	(1)<Business Channel>.
System	Low G.	(1)<Data Entity>, <Data Subject Area>, <Data Entity Relationship>. (2)<Data Object>, <Data Input>, <Data Output>. (3):<Entity>. (7),(8):<Class>. (11):<Class>. (20):<Entity>.	(1)<System Plan>,<Object Rule>.	(1)<Work Group>. (3):<Actor>, <Control>. (10):<System Actor>. (20):<Actor>, <Control>.	(1)<System Processing Cycle>. (2)<Message Flow>.	(1)<Processing Site>. (3)<Component>.
	High G.	(1)<Data Attribute>, <Data Identifier>, <Data Domain>. (2)<Data Object>, <Data Input>, <Data Output>. (3):<Entity>. (7):<Domain>, <Identification>, <Attribute>, < > (11):<Class>. (20):<Entity>.	(1)<System Objective>, <Information Requirement>, <User View>,<Integrity Constraint>. (7),(8):<Constraint>, <Valuation>, <Precondition>	(1)<User Role>. (3):<Actor>,<Control>. (7),(8):<Role>. (20):<Actor>,<Control>.	(1)<S.Event> (2)<Start Event>, <Intermediate Event>, <End Event>. (3)<Life Line>, <Event>, <Sequence number>. (7)(8):<Constraint>, <Event> (10)<Time Operators>	(1)<Communication Link> (3)<Package>



Tabla 3.6. Clasificación Taxonómica de la Actividad para Datos, Motivación, Personas, Tiempo y Ubicación, Capa Colaboración

Activity Taxonomic Classification, Collaboration Layer						
Business		Data	Goal	Who	When	Where
	Low G.					
High G.						
System		Data	Goal	Who	When	Where
	Low Granularity	(4):<Object>, <Notification>, <Data Dependency>. (20):<Object>, <Notification>.	(9):<Interaction Protocol>	(4):<Group>, <Work Team>, <Relationship>. (9):<Actor>, <Capacity> <Group> (13):<Actor>, <Group>, <Individual>, <User>, <Role>, <Agent>. <Relationship>. (20):<Group>, <Work Team>, <Relationship>.		
High G.	(9):<Information Object>	(9):<Law>		(4):<Temporal Dependency>, <Deadline Dependency>, <End Dependency>, <Execution Dependency>. (9):<Event>		

Tabla 3.7. Clasificación Taxonómica de la Actividad para Datos, Motivación, Personas, Tiempo y Ubicación, Capa Interfaz de Usuario

Activity Taxonomic Classification, User Interface Layer						
Business		Data	Goal	Who	Where	When
	Low G.		(14):<Concept/Set>	(14):<Post-condition>	(14):<Actor>. (15):<Job>, <Job Group>, <Organizational Unit>.	(15):<Agenda>.
H. G.						
System		Data	Goal	Who	Where	When
	Low Granularity	(10):<Input Element>. (12):<Artifact>. (14):<Gate> (16):<Class>. (13):<Input>.		(10):<User>. (4):<Actor>, <Role>, <Software Agent>, <Relationship>. (12):<User Actor>, <User Role>. (14):<User>, <System>.. (18):<Role>. (20):<Actor>, <Role>, <SystemActor>, <Software Agent>, <Relationship>.		
H.G.		(16):<Class> (18):<Class>				

La clasificación resultante permite concluir que la mayoría de propuestas aportan elementos de modelado para la capa funcionalidad. Igualmente, se observa que las celdas menos cubiertas son las que describen Tiempo, Ubicación y Motivación. Es decir, la mayoría de las propuestas se centran en especificar la actividad teniendo en cuenta información relacionada con los Datos, las Personas y la Función.

3.4.4 Clasificadores complementarios para cada celda de la TxA

En esta sección se presenta la forma como se enriquece la definición de la TxA de acuerdo al análisis para la clasificación complementaria, paso 4 de acuerdo con el mapa de ruta presentado en la Figura 3.3.

El modelado de sistemas interactivos debe llevarse a cabo de manera integral. Es decir, el modelado de la actividad debe relacionarse con los demás modelos que representan el sistema interactivo completo, Modelos de la Organización, de los Datos, de los Lugares, etc. El mismo metamodelo propuesto por Zachman (Sowa et al, 1992) lo expresa de esa manera. En este sentido, se analiza la forma como se aborda la clasificación de la actividad, esta vez desde los clasificadores de la taxonomía de Zachman. Los clasificadores "Data", "People", "Motivation", "Why", "Time" se encuentran fuera de la actividad "Function", y por tanto la información clasificada por ellos. Se realiza dicha clasificación, para posteriormente utilizar la información clasificada dentro de la categoría de la función, como elementos de relación con las demás columnas. Las figuras 3.11 y 3.12 muestran el metamodelo de Zachman de acuerdo a los conceptos que se ubican en cada columna. Los elementos "Business Resource" y "Business Product Service" se clasifican en todas las columnas porque se considera que un "Recurso" y un "Servicio" son productos que contienen información relacionada con la totalidad de las celdas. Igualmente, sucede con los elementos "User View" e "Information Requirement".

Por otro lado, lo que se pretende mostrar es que en realidad a través de la actividad se modelan todas las celdas de la Taxonomía de Zachman. La Figura 3.13 muestra una fracción del metamodelo, donde desde la Función se tienen referencias a los demás elementos. Las relaciones son la información de referencia. La Figura 3.14 muestra el metamodelo con todas las relaciones entre columnas. En este punto interesa recordar que uno de los usos principales de la Taxonomía de la Actividad que se propone en este trabajo, es el de definir lenguajes de modelado para sistemas interactivos. En este sentido, como lo expresa Giraldo (Giraldo, 2010) la Taxonomía debe proporcionar un conjunto de conceptos dentro de una misma celda, para así representar modelos sobre una vista y una perspectiva en particular. Además, los conceptos que se clasifican dentro de la Taxonomía deben tener *Independencia Taxonómica*, es decir, que la instanciación de un concepto no implica la instanciación de otro. Por ejemplo, en la Figura 3.15 los conceptos *OrganizationalUnit* y *Objectives* se clasifican en celdas distintas, *People* y *Motivation*, respectivamente, sin embargo, una vez que se instancia un *OrganizationalUnit* se instancia también un *Objective* por tener una relación estructural predefinida "atributo de".

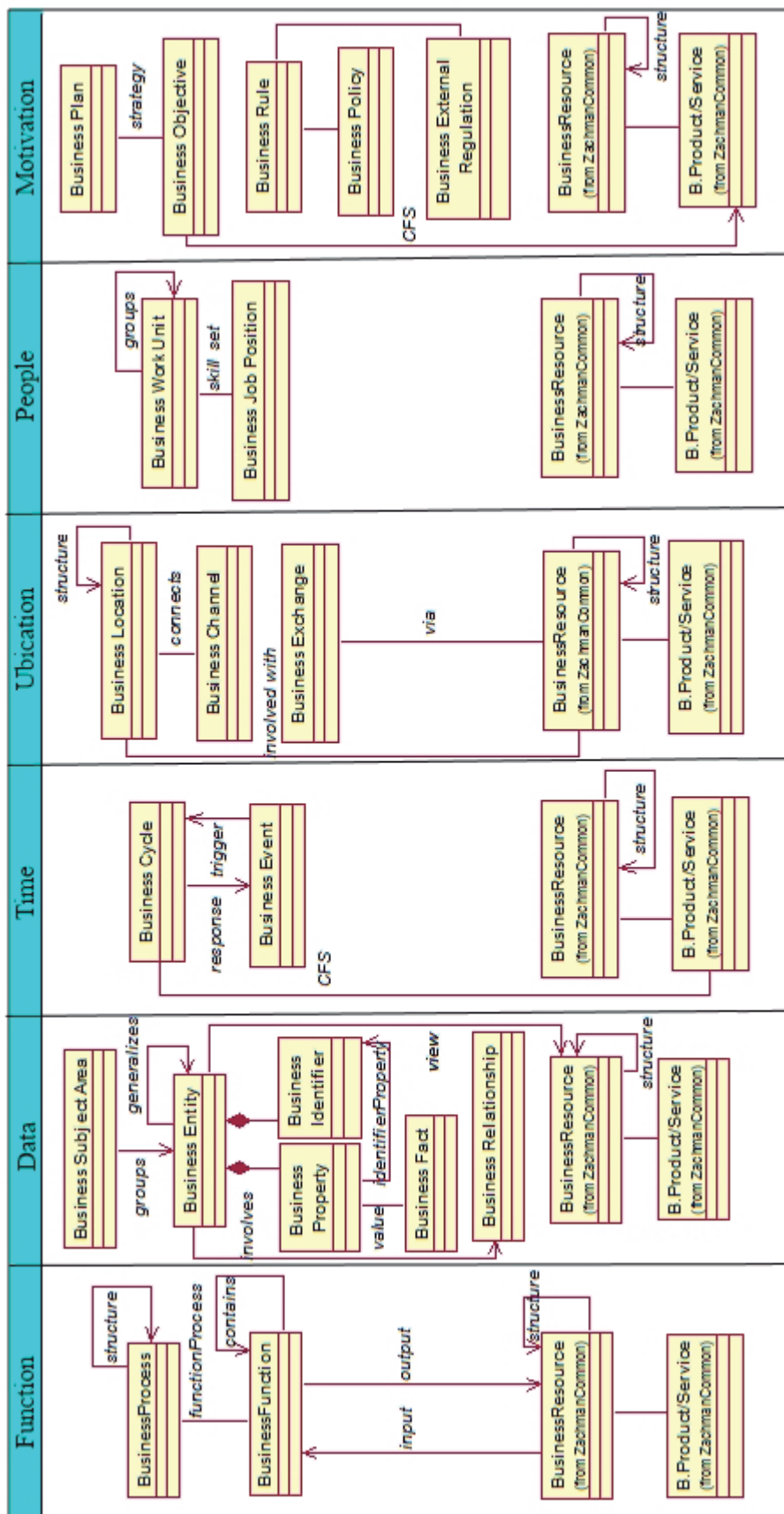
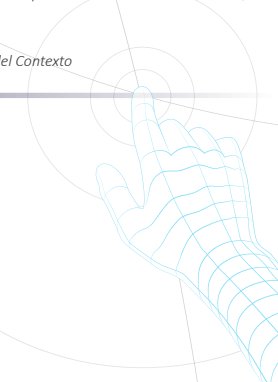


Figura 3.1.1. Metamodelo detallado de la taxonomía propuesta por Zachman, Nivel de Negocio



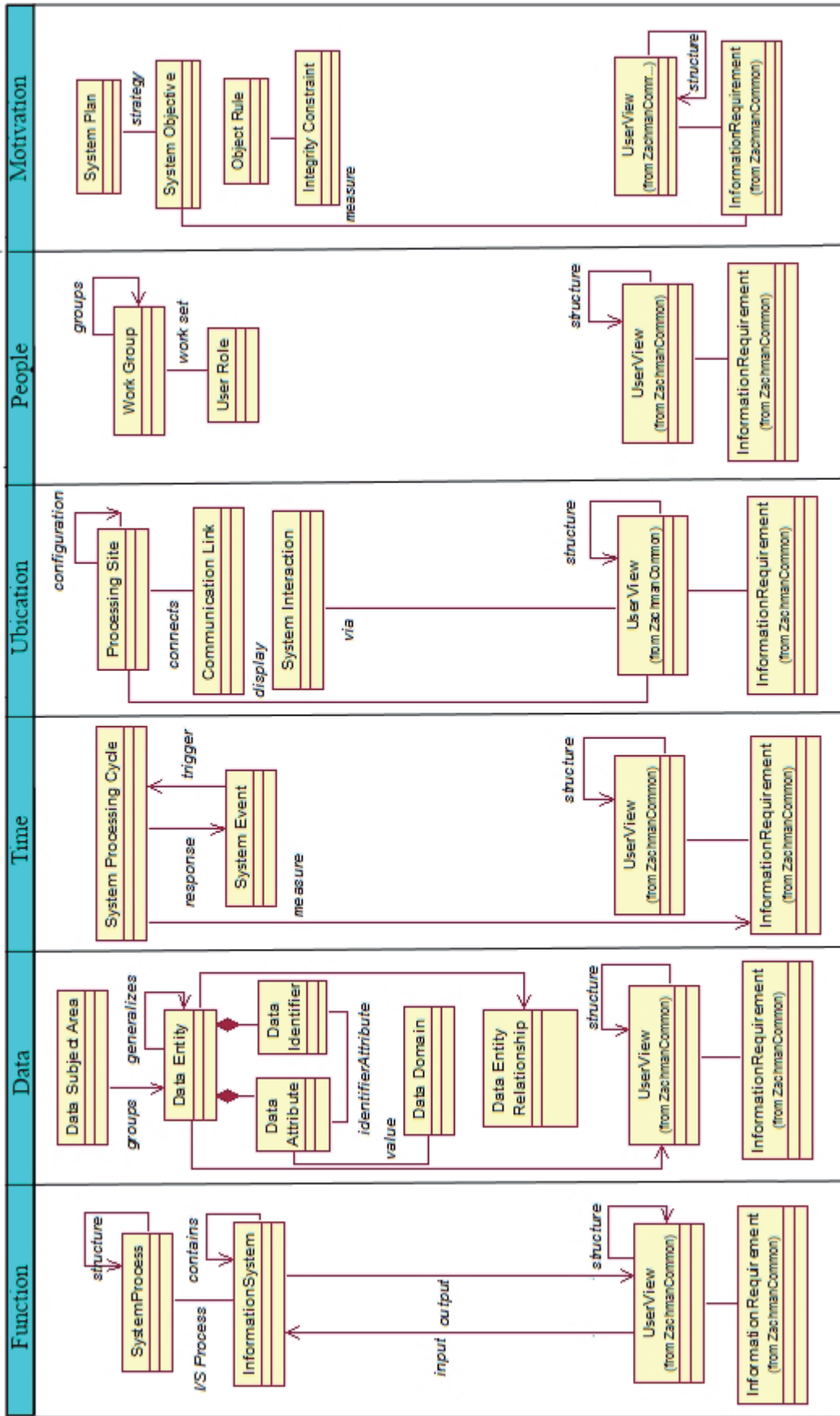


Figura 3.12. Metamodelo detallado de la taxonomía propuesta por Zachhman, Nivel de Sistema

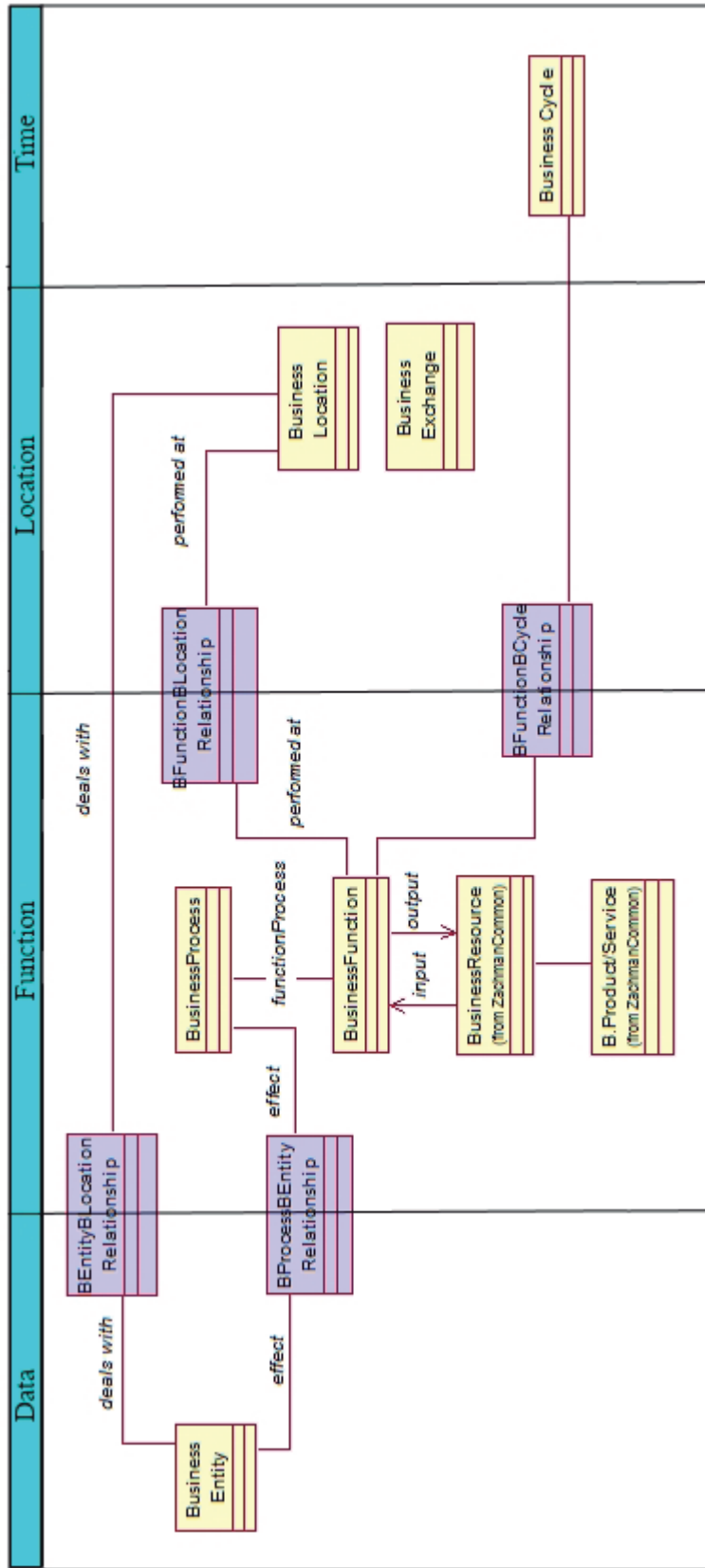


Figura 3.13. Fracción del metamodelo de Zachman. Relaciones entre columnas a partir de la Función



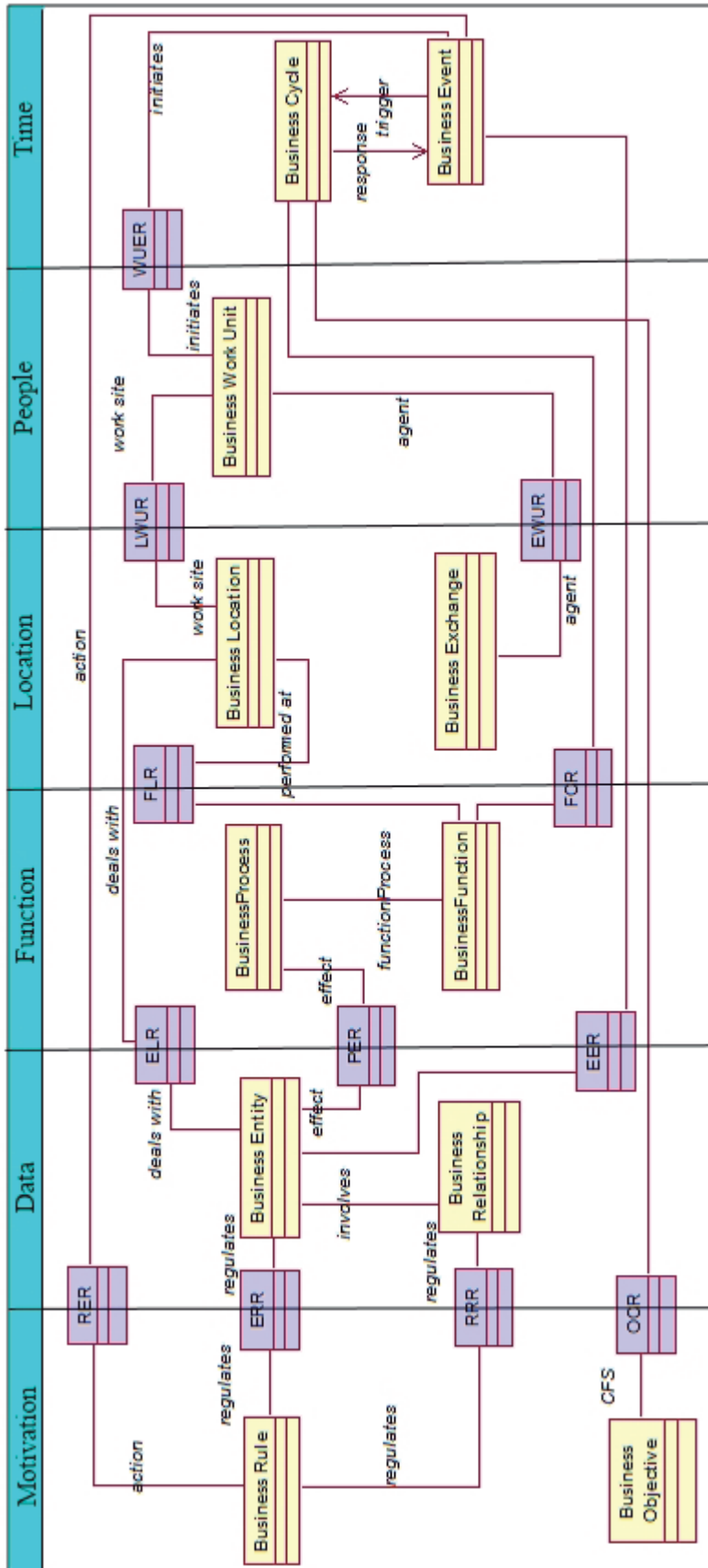


Figura 3.14. Metamodelo de Zachman. Relaciones entre columnas a partir de la Función

La falta de independencia entre estos dos conceptos limita la definición de una sintaxis concreta para especificar un diagrama de objetivos que esté separado del modelo de tareas.

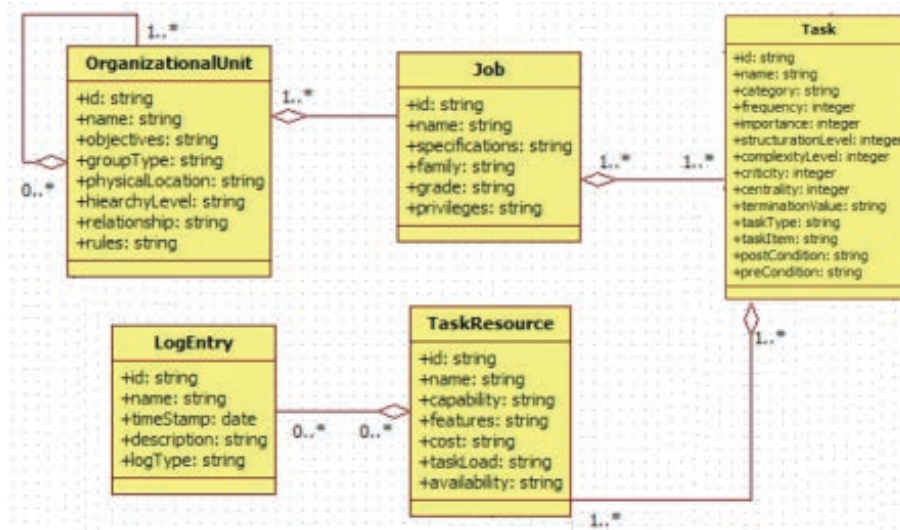


Figura 3.15. Fracción del metamodelo conceptual de la propuesta metodológica para el desarrollo de IUs a partir de Workflows (Guerrero, 2010)

En la misma línea de la Independencia Taxonómica, una característica que se destaca del lenguaje UML y que puede ser observada mediante el método de clasificación propuesto, es que cada uno de sus diagramas se centra en una celda en particular. Posiblemente, esta independencia entre los conceptos que forman su marco conceptual es una de las características que lo han hecho tan popular. La Figura 3.14 muestra una fracción del Metamodelo para el Diagrama de Casos de Uso. Se observa que el concepto *UseCase* es independiente del concepto *Actor*. Por otro lado, se sabe que un Diagrama de Casos de Uso necesariamente debe tener actores, un Diagrama de Actividad debe tener Swimlanes que representan los actores, una Actividad necesita referenciar los datos que afecta.

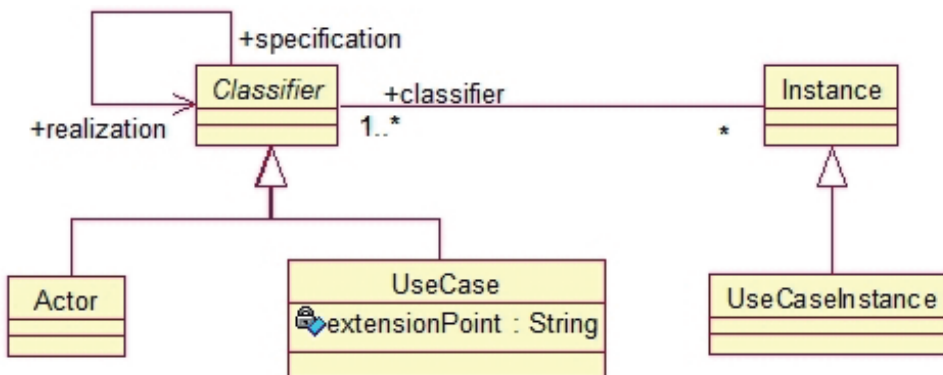
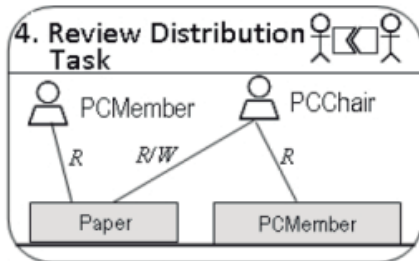


Figura 3.16. Fracción del metamodelo de UML v1.1. para Casos de Uso (OMG, 1999)

Si lo que se desea es tener un lenguaje con expresividad alta en Actividad, se requieren elementos de modelado que permitan visualizar la mayor cantidad de información que se captura a través de los diagramas, como por ejemplo, en el elemento "Tarea Cooperativa" propuesto por CIAF (Figura 3.17).



En el elemento "Tarea Cooperativa" propuesto por CIAF es posible visualizar no sólo el nombre y el tipo de tarea, sino también los responsables de ejecutar la tarea, los datos que manipulan y los tipos de acceso a

Figura 3.17. Representación de una tarea cooperativa en CIAF (Giraldo, 2010)

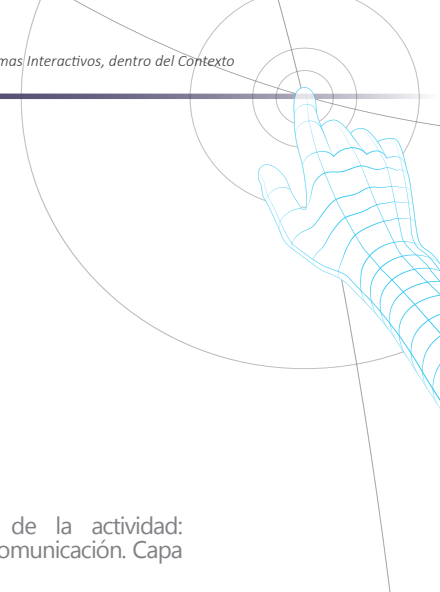
Aunque cada elemento se clasifica en una columna distinta en la Taxonomía, desde la Función debe permitirse tener las referencias a dichos elementos. Referencias que permitan relacionar los distintos modelos. De esta forma la Actividad se convierte en el elemento integrador, de orquestación a través del cual interactúan los demás elementos que permiten especificar un sistema interactivo y capturar la mayor cantidad de información para describir completamente un sistema sin que dicha información exista de forma aislada.

Según el metamodelo detallado de la Taxonomía propuesta por Zachman (Figuras 3.11 y 3.12), se deben adicionar dos clasificadores en el nivel principal de la Taxonomía, que son **Tiempo** y **Ubicación**. Por otro lado, se han identificado un conjunto de clasificadores que podrían ser agrupados en "tareas que involucran comunicación" (Tablas 3.8 a 3.10), lo que se cataloga en la notación CTT (Paternò, 2004) como Tarea Interactiva y en la propuesta "Model Driven Software Development Framework" (Cubillo, 2011), como Eventos Comunicativos. Estos clasificadores se ubican entonces en la celda **Función**.

Los clasificadores complementarios permiten tener una vista de Clasificadores y Sub-Clasificadores para la Taxonomía de la Actividad, como se muestra en la Figura 3.18.

El conjunto de clasificadores proporcionados por la TxA permiten definir completamente un sistema en torno a una serie de capas, aspectos o facetas (colaboración, funcionalidad, interacción, etc.). Dichos clasificadores representarían cada aspecto del modelado de la actividad.

La definición de la TxA contiene también un proceso de clasificación y un conjunto de reglas (Unterkaalmsteiner et al, 2014), (Giraldo, 2010) para controlar la integridad, la unicidad, la consistencia y la recursividad de la información especificada, como se describe en la sección 3.4.4.1.



Activity Taxonomic Classification for Communication, Functionality Layer		
Business	Low Granularity	(1)<Business Exchange>. (2)<Send Task>, <Receive Task>, <Choreography Task>, <Message>, <Conversation>. (3):<Relationship>. (10):<Relationship>. (11):<Communicative Event>. (20):<Relationship>.
	High G.	(1)<Business Exchange>. (2)<Sub-Choreography Task>, <Sub-Conversation>. (3):<Relationship>. (20):<Relationship>.
System	Low G.	(1)<System Interaction>
	High G.	(1)<System Interaction>. (3):<Interfaz>. (7),(8):<Interface>. (10):<Communicate>. (20):<Interfaz>.

Tabla 3.8. Clasificación taxonómica de la actividad: Elementos de modelado que denotan comunicación. Capa Funcionalidad.

Tabla 3.9. Clasificación taxonómica de la actividad: Elementos de modelado que denotan comunicación. Capa Interfaz de Usuario.

Activity Taxonomic Classification, for Communication, User Interface Layer		
Business	Low G.	(10):<Interaction Space>,<Communicate>, <Navigate>.
	H.G.	
System	Low Granularity	(10):<Task>,<Communicate>, <Temporal Relationship>. (12):<Task>. (14):<Super-Interactor>. (6):<Interactive Individual Abstract Task>, <User Individual Input>, <User Individual Output>, <User Individual IO>.
	High Granularity	(10):<Action>. (4),(5):<CTT Interaction Task>. (13),(5):< CTT Interaction Task >. (14):<Sub-Interactor>. (15),(5):< CTT Interaction Task >. (16),(5):< CTT Interaction Task >. (17),(5):< CTT Interaction Task >,<State>, <Transition>,<Event>. (6):<Interactive Individual Input>, <System Input>,<System Output>, <Interactive Individual Output>, <Interactive Individual IO>, <System IO>. (20),(5):<CTT Interaction Task>.

Tabla 3.10. Clasificación taxonómica de la actividad: Elementos de modelado que denotan comunicación. Capa Colaboración

Activity Taxonomic Classification, for Communication, Collaboration Layer		
Business	Low G.	Communicative Task
	High G.	
System	Low Granularity	(4):<Notification Dependency> (13):<Co-interaction>. (6):<Interactive Cooperative Abstract Task>, <User Cooperative Input>, <User Cooperative Output>, <User Cooperative IO>. (9):<Cooperative Task>
	High Granularity	(6):<Interactive Cooperative Input>, <Interactive Cooperative Output>, <Interactive Cooperative IO>, <Interactive Individual Group Task>, <Interactive Cooperative Group Task>.

Estructura Taxonómica de la Actividad, una sola Capa												
Negocio	Función					Dato	Meta		Persona		Tiempo	Ubicación
	Base		Soporte		Gestión		O	Re	Cl	Wo		
	C	I-A	C	I-A	W							
G. Baja												
G. Alta												
Sistema	Función					Dato	Meta		Persona		Tiempo	Ubicación
	Base		Soporte		Gestión		O	Re	Us	Si		
	U	I	U	I	S							
G. Baja												
G. Alta												
C: Tarea que hace el Cliente			I-A: Inter-Acción			W: Tarea que hace el Worker						
O: Objetivo			Re: Regla			Cl: Cliente						
Wo: Trabajador (Worker)			Us: Usuario			Si: Sistema						
S: Tarea que hace el Sistema			I: Interacción			U: Tarea que hace el Usuario						
Wo: Trabajador (Worker)			Us: Usuario			Si: Sistema						
S: Tarea que hace el Sistema			I: Interacción			U: Tarea que hace el Usuario						
Wo: Trabajador (Worker)			Us: Usuario			Si: Sistema						
S: Tarea que hace el Sistema			I: Interacción			U: Tarea que hace el Usuario						

Figura 3.18. Estructura taxonómica de la actividad

3.4.4.1 Proceso de clasificación y reglas definidas para la TxA

El método de clasificación explica cómo se clasifican los objetos bajo estudio dentro de la taxonomía, de forma repetida y sin ambigüedades. Se define el proceso (Tabla 3.11) en el cual cada paso responde una pregunta específica. Cada pregunta se formula de acuerdo a si se valora un caso de estudio (1) o una metodología (2) de desarrollo para sistemas interactivos.

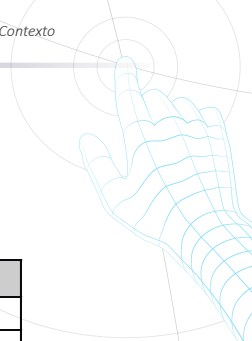


Tabla 3.11. Proceso de clasificación de la TxA

Paso	Pregunta	Tema
1.	¿El estudio abarca aspectos del desarrollo de sistemas interactivos?	Relevancia
2.	(1): ¿Cuáles son los procesos que especifica? (2): ¿En qué aspecto del desarrollo de sistemas interactivos se enfoca?	Caracterización básica
3.	(1): ¿En qué niveles de abstracción (negocio, sistema) se encuentra especificada la información? (2): ¿En qué niveles de abstracción (negocio, sistema) se proveen elementos de modelado?	Niveles de Abstracción
4.	(1): ¿Se diferencian los tipos de actividad (base, soporte, gestión)? (2): ¿Qué elementos de modelado provee para clasificar los tipos de actividad (base, soporte, gestión)?	Tipo de Actividad
5.	(1): ¿Cómo clasifica las intenciones del actor de negocio, las intenciones del worker y sus inter-acciones separadamente? (2): ¿Qué elementos de modelado provee para clasificar las intenciones del usuario, las intenciones del worker y sus inter-acciones separadamente?	Especificación a nivel de negocio (si aplica). Si no aplica, pasar a la especificación a nivel de sistema (paso 10)
6.	(1): ¿Cómo se diferencia el nivel de granularidad en el que se especifican las actividades a nivel de negocio? (2): ¿Qué elementos de modelado provee para diferenciar el nivel de granularidad en el que se especifican las actividades a nivel de negocio?	Nivel de granularidad a nivel de negocio
7.	(1): ¿Cómo se diferencian los roles que intervienen en las actividades a nivel de negocio? (2): ¿Qué elementos de modelado provee para diferenciar los roles que intervienen en las actividades a nivel de negocio?	Roles a nivel de negocio
8.	(1): ¿Cómo se clasifican los objetivos a los cuales contribuyen la ejecución de las actividades, a nivel de negocio? (2): ¿Qué elementos de modelado provee para clasificar los objetivos a los cuales contribuye la ejecución de las actividades a nivel de negocio?	Metas-Objetivos a nivel de negocio
9.	(1): ¿Cómo se clasifican los datos relacionados con la ejecución de las actividades, a nivel de negocio? (2): ¿Qué elementos de modelado provee para clasificar los datos relacionados con la ejecución de las actividades a nivel de negocio?	Datos a nivel de negocio
10.	(1): ¿Cómo se clasifican las intenciones del actor de sistema, las responsabilidades del sistema y sus interacciones separadamente? (2): ¿Qué elementos de modelado provee para clasificar las intenciones del actor de sistema, las responsabilidades del sistema y sus interacciones separadamente?	Especificación a nivel de sistema.
11.	(1): ¿Cómo se diferencia el nivel de granularidad en el que se especifican las actividades a nivel de sistema? (2): ¿Qué elementos de modelado provee para diferenciar el nivel de granularidad en el que se especifican las actividades a nivel de sistema?	Nivel de granularidad a nivel de sistema
12.	(1): ¿Cómo se diferencian los roles que intervienen en las actividades a nivel de sistema? (2): ¿Qué elementos de modelado provee para diferenciar los roles que intervienen en las actividades a nivel de sistema?	Roles a nivel de sistema
13.	(1): ¿Cómo se clasifican los objetivos a los cuales contribuyen la ejecución de las actividades, a nivel de sistema? (2): ¿Qué elementos de modelado provee para clasificar los objetivos a los cuales contribuye la ejecución de las actividades a nivel de sistema?	Metas-Objetivos a nivel de sistema
14.	(1): ¿Cómo se clasifican los datos relacionados con la ejecución de las actividades, a nivel de sistema? (2): ¿Qué elementos de modelado provee para clasificar los datos relacionados con la ejecución de las actividades a nivel de sistema?	Datos a nivel de sistema
15.	Discusión acerca de la Clasificación realizada	Conclusión



A continuación se describe la ejecución detallada del proceso:

- **Relevancia:**

La persona que aplica la taxonomía decide si el caso de estudio o la metodología de desarrollo califican para ser clasificado con la taxonomía. Su decisión está basada en dos criterios independientes:

- *Alcance:* Los estudios y metodologías candidatas deben considerar aspectos del desarrollo de sistemas interactivos, funcionalidad, desarrollo de la interfaz de usuario, colaboración, en general de la Interacción Humano Computador.
- *Nivel de especificación:* Es necesario contar con un reporte detallado de los estudios a analizar. Así se reduce el margen de interpretación al contestar las preguntas definidas para el proceso

- **Caracterización básica:**

El propósito es identificar aspectos básicos de los estudios a analizar. Para los casos de estudio se realiza una caracterización de los procesos que especifican a través de casos de uso, por ejemplo, así como de las metas de negocio a las que apuntan dichos procesos. Para las metodologías, se identifica qué aspecto del desarrollo de los sistemas interactivos se centran en desarrollar.

- **Desde Niveles de Abstracción hasta Datos:**

Se realiza el análisis de clasificación ubicando elementos de la actividad de acuerdo a los elementos clasificadores definidos en la Figura 3.18.

Finalmente, y una vez clasificada la información, se pasa a identificar qué clasificadores no han sido contemplados en la especificación, y es necesario incorporar. También es posible que se identifiquen clasificadores necesarios a incluir en la definición de la taxonomía.

Es importante resaltar que no toda la información que describe un sistema es necesaria para su incorporación en la Taxonomía. Solo es útil la información que se requiere que se transfiera al nivel de la tecnología.

La taxonomía también define una serie de reglas para controlar la integridad, la unicidad, la consistencia y la recursividad de la información especificada tal y como lo sugiere (Giraldo, 2010). En este sentido, aplican la mayoría de reglas definidas por la arquitectura de Zachman (Zachman, 1987).

- **Regla 1:** Las columnas Función, Dato, Meta y Quién no tienen orden. Todas son igualmente importantes.
- **Regla 2:** La estructura taxonómica interna de cada columna debe mantener el mismo orden.

- ⊙ **Regla 3:** Cada columna Función, Dato, Meta y Quién tiene un metamodelo base. Es base porque es el mismo para cada celda en la misma columna.
- ⊙ **Regla 4:** El metamodelo base de cada columna debe ser único. La unicidad es esencial para cualquier esquema de clasificación útil.
- ⊙ **Regla 5:** Cada fila Negocio, Sistema, representa una perspectiva distinta y única. Cada una trata con distintas restricciones del modelo.
- ⊙ **Regla 6:** La estructura taxonómica interna de cada fila debe mantener el mismo orden.
- ⊙ **Regla 7:** Cada celda es única, porque tiene un único modelo básico, que hace a cada columna única y pertenece a una sola perspectiva.
- ⊙ **Regla 8:** La composición o integración de todos los modelos de las celdas en una fila constituye un modelo completo desde la perspectiva de esa fila.
- ⊙ **Regla 9:** La lógica es recursiva.

Una vez definidos tanto el conjunto de clasificadores, como el método de clasificación y el conjunto de reglas para la TxA, se considera que la definición y seguimiento del mapa de ruta para la definición de la taxonomía, ha dado cumplimiento a las **Estrategias 4 y 5**, planteadas en la sección 1.2. Particularmente, el análisis realizado en la sección 3.4.4, acerca de la independencia taxonómica y la expresividad de las notaciones, ha dado cumplimiento a las **Estrategias 6 y 7**.

De esta forma es como se construyó la Taxonomía de la Actividad. Esta taxonomía garantiza que mediante un marco de referencia se puedan realizar los procesos de sincronía integración y extensión de artefactos para que sea posible la definición de herramientas, métodos y notaciones por medio de un roadmap.

La forma de validar la taxonomía se realiza mediante la definición y la aplicación de usos de la TxA para apoyar el desarrollo de los sistemas interactivos. En el capítulo 4 se describen los usos para la TxA y en el capítulo 5 se describen las aplicaciones para dichos usos. De esta forma, la taxonomía expone todo su potencial.



A person's hands are shown typing on a laptop keyboard. The image is overlaid with a semi-transparent green filter. In the upper right corner, there is a large, stylized number '4' composed of dark green and black geometric shapes. The laptop screen displays a line graph with multiple colored lines (green, orange, blue) and a grid background. The person's left hand is wearing a watch.

USOS DE LA TAXONOMÍA

CAPÍTULO CUARTO

4. USOS DE LA TAXONOMÍA

4.1 VALORACIÓN DE ARTEFACTOS DE LA INGENIERÍA DE SOFTWARE

4.1.1 Validación de la usabilidad en casos de uso mediante la taxonomía de la actividad

4.2 EVALUACIÓN DE LENGUAJES DE MODELADO

4.2.1 Evaluación de elementos de modelado en el desarrollo de sistemas interactivos

4.2.2 Niveles de abstracción y modelado de la información

4.3 DEFINICIÓN DE MARCOS DE DESARROLLO DE SISTEMAS INTERACTIVOS

4.3.1 Contexto desde los puntos de vista de la Ingeniería de Software y HCI

4.3.2 Componentes del roadmap

CAPÍTULO CUARTO

4. USOS DE LA TAXONOMÍA

Las taxonomías de objetos y conceptos se han convertido sin duda en una herramienta científica básica, como se resume en (Unterkalmsteiner et al, 2014). Los sistemas de clasificación permiten la visualización de la teoría de una manera útil y servir, de forma similar a las teorías, como conductores para la investigación (Kwasnik, 1992). Particularmente, en el ámbito de la Ingeniería de Software, las taxonomías se convierten entonces en un elemento esencial para documentar teorías que acumulan el conocimiento (Sjoberg et al, 2007).

En este capítulo se describen los usos previstos para la Taxonomía de la Actividad propuesta en esta tesis, todo esto dentro del contexto de la HCI, para luego describir la aplicación y valoración por cada uso mediante la definición de un caso específico. La figura 4.1 presenta los usos de la TxA en los que se ha enfocado esta tesis: 1. Valoración de artefactos de la Ingeniería de Software; 2. Evaluación de lenguajes de modelado; y 3. Definición de marcos de desarrollo de sistemas interactivos.

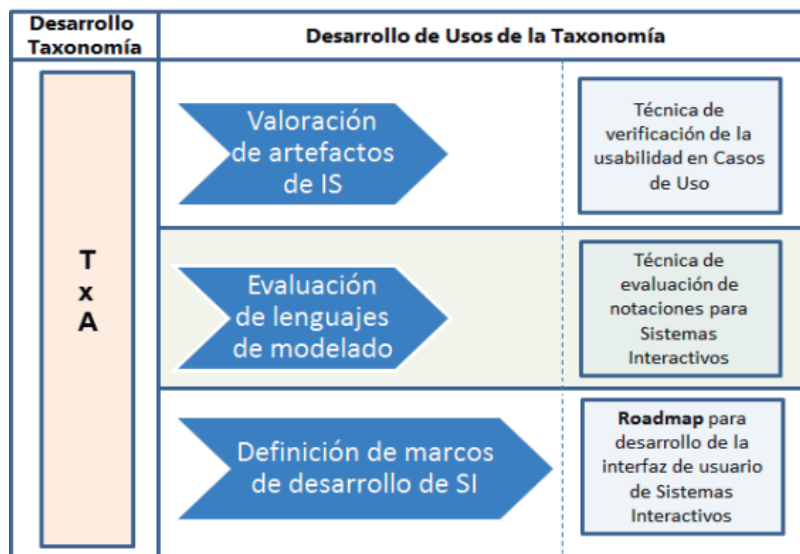


Figura 4.1. Usos definidos para la taxonomía de la actividad propuesta

La validación de la TxA, mediante los usos definidos en este capítulo, permite dar cumplimiento a las **Estrategias 17,18 y 19**, planteadas en la Sección 1.2.

4.1 VALORACIÓN DE ARTEFACTOS DE LA INGENIERÍA DE SOFTWARE

Desde el punto de vista de la valoración de artefactos utilizados en la Ingeniería de Software, se toma la TxA como referencia para realizar una evaluación de la Usabilidad de un Sistema Interactivo, a partir de la especificación de sus Casos de Uso. Específicamente para esta investigación, nos interesan las siguientes características de los casos de uso: i) normalmente se utilizan en las etapas tempranas del desarrollo de un sistema interactivo y ii) forman parte de las principales metodologías de desarrollo de software (Hornbæk et al, 2007). Adicionalmente, se parte de los Casos de Uso, porque se han definido como una excelente y poderosa forma de definir los requerimientos de comportamiento del software (Gottesdiener, 2002), porque permiten definir un proyecto y establecer un lenguaje común entre stakeholders. Todo esto permite considerarlos como una base potencialmente adecuada para realizar evaluación de usabilidad de un producto software durante las etapas tempranas de su desarrollo.

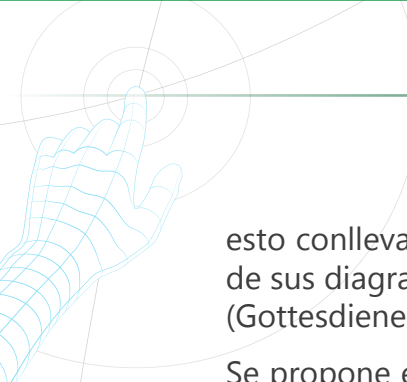
4.1.1 Validación de la usabilidad en Casos de Uso mediante la taxonomía de la actividad

La Ingeniería de Software se ha centrado principalmente, en generar productos completamente funcionales, es decir, se ha enfocado en el aspecto de la funcionalidad, dejando un poco de lado otros atributos inherentes al desarrollo de sistemas interactivos, como la colaboración, la usabilidad, entre otros. La respuesta de la comunidad investigativa ante este enfoque netamente funcional, han sido los Casos de Uso e Historias de Usuario.

Los casos de uso, tradicionalmente, han sido utilizados por equipos de desarrollo de software como una herramienta técnica y administrativa. Desde cada una de estas perspectivas, es necesario tener un conjunto de atributos a gestionar, ya que la información que se captura para el desarrollo de un producto software, contiene información tanto del sistema, como de la organización. Desde el punto de vista de la HCI, el interés se centra más en capturar el modelo mental de los usuarios y en otros aspectos relacionados a la psicología cognitiva y al factor humano. Si toda esta información, tanto desde la Ingeniería de Software, como desde la HCI, pretende ser capturada a través de los Casos de Uso, probablemente, lo que resulta es un aumento significativo de la complejidad de su especificación, lo cual genera cierto nivel de confusión a la hora de utilizarlos.

Las discusiones en la literatura y la experiencia adquirida en evaluación de la usabilidad y en desarrollo de interfaces de usuario, indican que al parecer existe un problema relacionado con la especificación de los Casos de Uso, más que todo en lo que tiene que ver con la separación de niveles de abstracción. Resulta confuso modelar con Caso de Uso cuando se mezclan *Workers*, *Entidades* y *Actores de Negocio*, *Realizaciones*, *Reglas*, *Acciones*, componentes que hacen parte de los *Casos de Uso de Negocio*, con especificaciones relacionadas con la funcionalidad a *nivel del Sistema Interactivo*. Todo





esto conlleva a un aumento de la complejidad o mal uso tanto al nivel del modelado de sus diagramas como a nivel de su especificación, "abuse cases", como lo denomina (Gottesdiener, 2002).

Se propone entonces la separación de elementos de negocio y técnicos, al momento de elaborar Casos de Uso, usando la TxA. Se presenta la TxA en el Modelado con Casos de Uso para reducir su complejidad y principalmente para incrementar su efectividad y utilidad en lo que tiene que ver con la Interacción Humano Computador.

Se propone estudiar los clasificadores o tokens que componen la especificación de los Casos de Uso y cómo se organizan en torno a la Taxonomía de la Actividad (TxA). La TxA estudia los clasificadores que hacen parte de las especificaciones de Casos de Uso desde distintos puntos de vista, no sólo desde el aspecto de la funcionalidad, sino también desde la usabilidad y del diseño de la interacción, para proveer un marco de organización de los clasificadores y analizar cómo a partir de ellos es posible crear artefactos, plantillas en el ámbito del desarrollo de sistemas interactivos. De esta forma podríamos determinar por ejemplo, si la funcionalidad a nivel de negocio, debe ser especificada en una plantilla distinta a la funcionalidad a nivel de sistema.

Esta propuesta no pretende que las empresas desarrolladoras de software incorporen de lleno el diseño centrado en el usuario en sus procesos, ni cambiar del todo los Casos de Uso. Lo que se pretende es proporcionar un conjunto de técnicas y artefactos que incorporen aspectos de HCI en el desarrollo de software.

4.1.1.1 Evaluación de la usabilidad y evaluación dirigida a los Casos de Uso

La literatura relacionada con la evaluación dirigida a los casos de uso discute una variedad de checklist, técnicas (Anda et al, 2002), guías (Jagielska et al, 2006), (Cockburn, 2000) heurísticas, métricas (Bernárdez et al, 2004), identificación y solución de problemas (Graham, 1996), orientados a mejorar la calidad de los modelos de casos de uso (Jacobson, 1992). Se argumenta que el incremento en la calidad de dichos modelos, tiene un impacto positivo directo en la calidad de los productos software que especifican. Particularmente, en estos trabajos se observa que no se enfocan en un atributo de calidad en especial.

Por otro lado, en la literatura relacionada con la evaluación de la usabilidad se argumenta que se realiza a partir de prototipos o sistemas (Rosson et al, 2002). Adicionalmente, se encuentran diferentes productos de diseño que pueden ser evaluados en etapas tempranas del proceso de desarrollo como storyboards, prototipos y especificaciones de interfaces (Granollers, 2004). En particular para los casos de uso, algunos autores argumentan que proveen una conexión fuerte entre el desarrollo de software y el diseño de las interfaces de usuario (Constantine et al, 1999), (Seffah et al, 2001). Ferré (Ferré et al, 2001) y Giraldo (Giraldo, 2010) por ejemplo, sustentan que los casos de uso ofrecen un buen punto de partida para integrar técnicas de usabilidad dentro de los procesos de desarrollo de software. Concretamente, Sutcliffe (Sutcliffe, 2014) trata sobre la integración de la HCI con la

Ingeniería de Requisitos, pero como asegura el mismo autor, aún no se adoptan procedimientos formales para chequear o probar los artefactos resultantes del diseño de software.

La evaluación de la usabilidad a partir de los casos de uso, es abordada por (Hornbæk et al, 2007), proponiendo un método que se basa en un conjunto de heurísticas definidas por Nielsen (Nielsen, 1994) para inspeccionar la especificación de los casos de uso. Realiza la validación comparando los problemas de usabilidad encontrados aplicando el método que propone y los problemas de usabilidad encontrados aplicando la técnica "Pensando en Voz Alta". Se observan otros trabajos en los que se propone hacer una evaluación temprana de artefactos como los escenarios de uso (Haynes et al, 2004), pero enfocados en aspectos distintos a la usabilidad o en mejorar atributos de calidad en general tal y como se mencionó anteriormente.

4.1.1.2 Especificación de la Actividad a través de Casos de Uso

Desde el punto de vista del desarrollo de software, la actividad ha sido muy discutida. Por ejemplo en (Nunes, 2001), se asegura que los Casos de Uso son deficientes para soportar las actividades centradas en el usuario, son imprecisos en su definición, son inútiles, no proveen una forma eficiente para soportar los aspectos de usabilidad del sistema interactivo y no son una forma generalizada de modelado de tareas. En (Lozano, 2001) se observa cómo se critica el lenguaje UML por no abordar el modelado de la interfaz de usuario, igualmente se critican los casos de uso por no permitir la captura de los requisitos no funcionales y se dice además que no permiten capturar las necesidades del usuario final de una aplicación informática. A partir de estas afirmaciones surgen preguntas como: los casos de uso son ¿buenos, malos, útiles?, ¿qué información capturan?, ¿qué información les falta por capturar?, etc. Discusiones y preguntas como éstas justifican un análisis del Modelado de la Actividad.

Se observa que los trabajos donde se realizan este tipo de discusiones no encuentran una base en la Clasificación del Modelado de la Actividad. Particularmente, para los CU, el problema radica en cómo se utilizan, no en cómo están especificados. Como lo expone (Cockburn, 2000), los CU capturan realmente los requerimientos cuando se describen de una forma apropiada. No proveen detalles sobre interfaces externas, formatos de datos, reglas de negocio, etc. Constituyen sólo una fracción de todos los requerimientos que deben ser capturados. Es decir, es imposible cubrir todos los aspectos en una sola especificación. Lo que se propone en este trabajo es aplicar la Taxonomía de la Actividad en la especificación de Sistemas Interactivos Clasificación de la Actividad en Niveles de Abstracción Negocio y Sistema, mediante la Taxonomía de la Actividad para disminuir la complejidad de los CU, haciendo una separación en CU de Negocio y CU de Sistema, de acuerdo con (Gottesdiner, 2002), (Ng, 2002), (Cockburn, 2000), (Langlands et al, 2009).

En la sección 5.1, se presentará con un caso de estudio, cómo la TxA soporta la evaluación de la usabilidad desde los casos de uso identificando los clasificadores de la actividad que se capturan a partir de las descripciones de la labor especificadas en los modelos de casos de uso.

4.2 EVALUACIÓN DE LENGUAJES DE MODELADO

En esta sección se presenta una revisión acerca de la evaluación de elementos de modelado en el desarrollo de sistemas interactivos. Adicionalmente, se describe la utilidad de la Taxonomía de la Actividad para evaluar elementos notacionales, desde el punto de vista de la captura de la información necesaria para el modelado de dichos sistemas.

4.2.1 Evaluación de Elementos de Modelado en el Desarrollo de Sistemas Interactivos

La literatura relacionada con la evaluación de elementos de modelado o notaciones utilizadas para modelar sistemas interactivos, en el área de HCI, es muy reducida. En trabajos como el de Haan (Haan et al, 1991), se analiza y comparan lenguajes de modelado enfocados a la HCI. La sintaxis concreta definida para estos lenguajes se encuentra especificada en forma textual, lo cual no aporta mucho al presente trabajo donde el interés se centra en notaciones especificadas de forma gráfica. Se encuentran también trabajos como el de Molina (Molina et al, 2009), donde se analiza y comparan notaciones para modelado conceptual de sistemas interactivos colaborativos. Dicho estudio se realiza con el fin de demostrar la necesidad de incorporar el aspecto de la colaboración en el modelado de este tipo de sistemas. Resulta interesante el listado de atributos analizados, por ejemplo si cada aproximación distingue entre lo cooperativo y lo colaborativo, o si tiene o no una metodología asociada. Se introduce entonces la propuesta metodológica CIAM (Molina et al, 2006), de gran interés para la presente investigación.

Por otro lado, los trabajos de Guerrero (Guerrero, 2010) y Limbourg (Limbourg y Vanderdonckt, 2004b) resultan muy interesantes desde la perspectiva de este trabajo, ya que realizan comparaciones entre notaciones, aunque sólo se tienen en cuenta notaciones para el modelado de tareas. Por esta razón los atributos que se comparan tienen que ver con la concurrencia, secuencialidad, respuesta del sistema, nivel operacional, etc.

Igualmente, Meixner (Meixner et al, 2011) y Balbo (Balbo et al, 2004) proponen taxonomías que permiten evaluar un conjunto de notaciones para modelado de tareas con el fin de seleccionar la mejor según las características del sistema que se desee modelar. Entre los criterios que definen estas propuestas para la evaluación de los modelos de tareas se encuentran: Grado de Formalidad, Integridad, Comunicabilidad, Editabilidad, Usabilidad, Adaptabilidad. Particularmente, Balbo propone el criterio de evaluación "*Poder Expresivo*", que relaciona la amplitud de lo que la notación evaluada puede expresar. Este criterio interesa para el presente trabajo, ya que uno de los atributos que se tienen en cuenta en la evaluación de las notaciones es "*Expresividad*".

En el Marco de Desarrollo de Sistemas Groupware Interactivos, CIAF, Giraldo (Giraldo, 2010) realiza una clasificación y comparación de propuestas metodológicas en relación a la notación, el proceso y el soporte mediante herramientas. En cuanto a cada notación, se clasifica "*Rigor Ontológico*", "*Mapping*" y "*Aspecto*". Se destaca que la comparación incluye propuestas no sólo para el modelado de tareas sino también de modelado de sistemas software en general. Aunque no se evalúa la calidad de los diagramas.



4.2.2 Niveles de abstracción y modelado de la información

Si bien lo que nos interesa en gran medida es el modelamiento de los sistemas, existen limitantes en la representación de la información que se modela en cuanto a niveles de abstracción, lenguaje y restricciones en dichos niveles de abstracción. P. ej., si tomamos la estructura básica de los niveles de transformación según el Framework de Zachman (Zachman, 2007-2011) para representar una arquitectura Empresarial, Conceptual, Lógica y Tecnológica, es posible diferenciar que cada nivel tiene sus propias restricciones (Tabla 4.1).

Tabla 4.1. Tipo de Restricción según el Nivel de Abstracción

Nivel de Abstracción	Tipo de Restricción
Conceptual (Negocio)	Restricciones Físicas
Lógico (Sistema)	Restricciones Lógicas
Tecnológico	Restricciones Tecnológicas

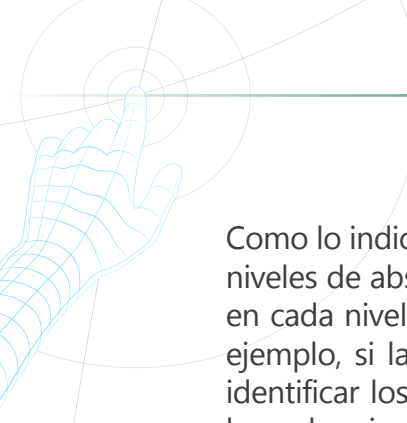
Es importante mencionar que los niveles de abstracción (rows) propuestos por el Framework de Zachman mapea los modelos CIM (Computation Independent Model), PIM (Plataform Independent Model) y PSM (Plataform Specific Model), definidos por la Arquitectura Dirigida por Modelos (MDA) (Frankel et al, 2003).

Si nos enfocamos en la Tabla 4.1, y nos preguntamos ¿Cuánto pesa una foto?, diferenciamos las restricciones que existen para cada nivel de abstracción:

- ⊙ A nivel de conceptual: la foto pesa en gramos (g), lo que pesa es el papel. ¿Cuántas fotos puedo almacenar?, (restricciones de espacio físico).
- ⊙ A nivel lógico: se define un arreglo con n elementos, fotos[1,···,n], (no hay restricciones lógicas).
- ⊙ A nivel tecnológico: la foto pesa en Kb, Mb,.., (restricciones de espacio en memoria).

Otro ejemplo podría ser: ¿Es posible comprar un boleto de avión para una fecha en el pasado?:

- ⊙ A nivel de conceptual: no es posible.
- ⊙ A nivel lógico: se define la Clase "Calendario" con los atributos necesarios para definir la fecha. (no hay restricciones lógicas).
- ⊙ A nivel tecnológico: el elemento "calendar" permite seleccionar una fecha en el pasado.



Como lo indica Zachman (Zachman, 2007-2011), se debe hacer un mapeo (*mapping*) entre niveles de abstracción a medida que se baja de nivel, es decir, el movimiento hacia abajo en cada nivel nada tiene que ver con el nivel de detalle, sino con la transformación. Por ejemplo, si la persona que atiende en una entidad bancaria en el rol de cajero puede identificar los billetes falsos (*nivel de negocio*), el cajero automático (ATM) debería poder hacer lo mismo (*nivel tecnológico*). Si no se identifica esta funcionalidad a nivel de negocio, seguramente tampoco se tendrá un *mapping* a nivel de la tecnología.

La funcionalidad, labor que se origina a nivel de negocio, se va heredando hasta el nivel de la tecnología, así que a nivel tecnológico es posible que falle un sistema si no se captura toda la información (labor, forma, datos) a nivel de negocio. Adicionalmente, de acuerdo con el concepto de "*Transformación*" mencionado por Zachman (Zachman, 2007-2011), no se deben hacer especificaciones que aborden más de un nivel de abstracción, es decir, en una misma especificación no se deben mezclar niveles de abstracción. Por ejemplo, no es posible que un atributo que defina la edad de una persona a nivel lógico esté relacionado con un atributo que defina la altura de una persona a nivel físico.

Los procesos que involucran captura de información en el modelado de sistemas interactivos utilizan comúnmente la abstracción de tipo funcional o de procedimiento y así se viene trabajando desde los años 70 (Liskov et al, 1975). Se observa que las abstracciones que se utilizan hoy en día para la especificación e implementación de sistemas software casi siempre cubren los clasificadores (celdas de información) ubicados en las vistas (*Interrogatives*) "*Qué*" y "*Cómo*" y un ejemplo claro de ello son los Casos de Uso. De hecho, lenguajes de programación como por ejemplo, el lenguaje Java (Corporation), definen las abstracciones "*for, if, while,..*" para el clasificador "*Cómo*", definen las abstracciones "*class, int,..*" para el clasificador "*Qué*". Pero no se encuentra un tipo de dato o componente léxico para describir, p. ej., una abstracción que corresponda con el clasificador "*Quién*", que represente un rol, o que represente tiempos, "*Cuándo*".

Es importante recordar que el propósito por el que son desarrollados los sistemas software es para alcanzar objetivos de negocio (Ng, 2002), es decir, la función primaria del software es automatizar la labor de los humanos. Labor que debe descomponerse para identificar las tareas que hace el humano, las que hace el sistema y las que hacen en conjunto. Así que, además de centrarnos en el aspecto de la funcionalidad, deberíamos centrarnos en otros tipos de actividades que el software debe soportar como el hecho de que las personas colaboren, se comuniquen, intercambien información, etc. Toda esta labor debe estar automatizada también para que el sistema la pueda hacer. Esta es la razón principal por la que se debe considerar un modelado de la actividad como base para generar software funcional. Es necesario entonces tener una clasificación de la actividad para volverla de tácita a explícita computacionalmente y soportar la automatización, como función primaria del software.

Finalmente, observamos que la estructura taxonómica de la actividad proporciona un conjunto de clasificadores definidos en cada una de sus celdas. Cada clasificador captura cierta información que se desea incorporar en el diseño para que el sistema interactivo funcione. Particularmente, el modelo de negocio en UML utiliza un solo clasificador (*Caso de Uso de Negocio*), es decir, no proporciona herramientas expresivas ni elementos diferenciadores.



Tampoco proporciona elementos de modelado para especificar las interacciones a nivel de negocio (inter-acciones) ni las interacciones a nivel de sistema. Si diferenciamos las tareas del actor de negocio de las tareas del worker (empresa), y de las tareas que hacen en conjunto (inter-acciones), se requiere una expresividad distinta para representar estos tres tipos de tareas, sus roles y sus relaciones.

El conjunto de clasificadores que proporciona la Estructura Taxonómica de la Actividad, se convierte en una base para la evaluación de elementos de modelado, analizando qué información se captura, cuánta labor, en qué nivel de abstracción siempre teniendo en cuenta el contexto y los objetivos del diseño. Con esta información se analiza la potencialidad que tiene cada notación para soportar herramientas que permitan la instrumentación y transformación de modelos a código ejecutable con Ingeniería Dirigida por Modelos.

En la sección 5.2, se presentará con un caso de estudio, cómo la TxA soporta la evaluación de elementos de modelado de un conjunto de notaciones utilizadas para representar la actividad en la especificación de sistemas interactivos.

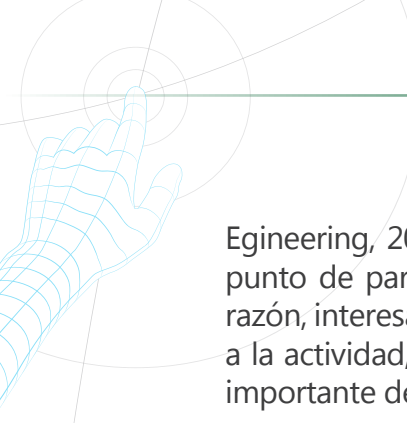
4.3 DEFINICIÓN DE MARCOS DE DESARROLLO DE SISTEMAS INTERACTIVOS

Esta sección describe un tercer uso, un poco más holístico, de la Taxonomía de la Actividad (TxA). Es la definición de métodos, herramientas y notaciones para proveer mapas de ruta (roadmaps) que interpreten aproximaciones metodológicas para el desarrollo de sistemas interactivos. Se entiende que la definición de un Marco de Desarrollo tiene el propósito de implementar una estructura de solución para una clase particular de problema dentro de un dominio específico (Giraldo, 2010).

4.3.1 Contexto desde los puntos de vista de la Ingeniería de Software y HCI

Los Sistemas de Información, así como los Sistemas Interactivos transaccionales son parte vital en las organizaciones modernas. Durante el desarrollo de este tipo de sistemas se capturan, generalmente a través de modelos, grandes volúmenes de información que contienen características y atributos tanto de la organización, como del sistema interactivo. La mayoría de profesionales en Desarrollo de Software comprenden que existe una división entre el lado del negocio, con sus requisitos, y entre el soporte que está siendo provisto para direccionar dichos requerimientos (García, 2010), (Trættemberg, 2002). Pero lo ideal no es que exista una división, sino más bien una trazabilidad entre estos dos niveles de abstracción, Negocio y Sistema.

Desde el punto de vista de la Ingeniería de Software (IS), lo que interesa es generar productos software completamente funcionales, es decir, se ha enfocado en el aspecto de la funcionalidad, dejando un poco de lado otros atributos inherentes al desarrollo de sistemas interactivos, como la colaboración, la usabilidad, entre otros (Trættemberg, 2002). Así, observamos que para la mayoría de propuestas metodológicas para desarrollo de software (IBM_Rational, 2003a), (Balduino, 2007), XP (Wells, 2013), ICONIX (ICONIX Software



Engineering, 2013), SCRUM (James, 2013), la especificación de la labor, la actividad, es el punto de partida hacia la implementación de cualquier producto de software. Por esta razón, interesa en gran medida para esta investigación los aspectos relacionados a la labor, a la actividad, porque es el elemento de cohesión, de orquestación, la característica más importante desde el punto de vista del desarrollo de software.

Desde el punto de vista de HCI, el interés se centra más en capturar el modelo mental de los usuarios y en otros aspectos relacionados a la psicología cognitiva y al factor humano (Granollers, 2004). Si toda esta información, tanto desde la Ingeniería de Software, como desde la HCI, pretende ser capturada para especificar y modelar productos software que además de ser funcionales, también sean usables, seguros, colaborativos, etc, se requieren entonces mecanismos y herramientas que lo soporten.

En este sentido, este trabajo sigue en la línea que define el desarrollo basado en Frameworks como una importante herramienta para el desarrollo de soluciones a problemas con características específicas (Ambler et al, 2005), y coincide con Giraldo (Giraldo, 2010) en que este tipo de desarrollo debe separar cada elemento de los modelos tanto a nivel de negocio, como de aplicación y tecnología, para así asegurar la agilidad y flexibilidad de los modelos del sistema.

4.3.2 Componentes del roadmap

En todo roadmap que soporte todo el ciclo de vida de un proyecto software se consideran indispensables tres elementos: notación que propone, metodología a seguir y herramientas suministradas (Giraldo, 2010). La metodología de desarrollo es soportada por un conjunto de mejores prácticas y unas bases de conocimiento que apoyan a los involucrados en la toma de decisiones de diseño. La notación, preferiblemente gráfica, permite la comunicación, la planificación y la descripción del sistema que se modela. Si además, ésta es formalizada, permite a su vez, el desarrollo de herramientas de edición y automatización.

El roadmap propuesto es una interpretación del Marco de Desarrollo de Sistemas Groupware Interactivos Basado en la Integración de Procesos y Notaciones (CIAF) definido por Giraldo (Giraldo, 2010). Específicamente, en la parte de la Interfaz de Usuario que se denomina TD-MBUID. Esta interpretación pasa por comprender por separado y en conjunto la notación utilizada, las herramientas utilizadas y el método como tal. Se utiliza entonces la TxA para hacer una adaptación tanto del método (TD-MBUID), como del lenguaje (CIAN+UML+UsiXML), y de la herramienta (CIAT). Esta adaptación, enfocada en el modelado y la ejecución de la actividad que soporta el diseño de la interfaz de usuario en sistemas interactivos.

En la sección 5.3 se describe con un caso de estudio el proceso de adaptación de los componentes del Marco de Desarrollo CIAF, a partir de la TxA. Además se muestra cómo la Estructura Taxonómica de la Actividad se convierte en una base para la definición de un conjunto de elementos de modelado diferenciadores en la especificación de sistemas interactivos; y posteriormente para la definición e implementación de herramientas que permitan la instrumentación y transformación de modelos de diálogo, modelos de datos, y modelos de tareas, a código ejecutable con Ingeniería Dirigida por Modelos.





**APLICACIÓN Y
VALORACIÓN
DE LOS USOS
DE LA TXA**

CAPÍTULO QUINTO

5. APLICACIÓN Y VALORACIÓN DE LOS USOS DE LA TXA

5.1 VALORACIÓN DE ARTEFACTOS DE LA INGENIERÍA DE SOFTWARE

5.1.1 Proceso de Incorporación

5.1.2 Equipo de expertos

5.1.3 Artefactos

5.1.4 Caso de estudio

5.2 EVALUACIÓN DE LENGUAJES DE MODELADO

5.2.1 Clasificación de elementos de modelado a partir de la TxA

5.2.2 Selección de elementos de modelado para la especificación de sistemas interactivos

5.3 DEFINICIÓN DE MARCOS DE DESARROLLO DE SISTEMAS INTERACTIVOS

5.3.1 Adaptación del método de desarrollo de la interfaz de usuario

5.3.2 Contenido metodológico detallado de la propuesta

5.3.3 Modificaciones a la notación usada en TD-MBUID y definición de algunos elementos de modelado

5.3.4 Modificaciones a la herramienta de software CIAT

5.3.5 Caso de estudio



CAPÍTULO QUINTO

5. APLICACIÓN Y VALORACIÓN DE LOS USOS DE LA TXA

El propósito de este capítulo es presentar, mediante casos de estudio, la validación de los usos que se describen para la TxA en el capítulo 4.

5.1 VALORACIÓN DE ARTEFACTOS DE LA INGENIERÍA DE SOFTWARE

En esta sección se presenta una aproximación metodológica para la Incorporación de HCI (Giraldo et al, 2014a) en procesos de desarrollo de software mediante la TxA. Dicha incorporación se orienta específicamente a la evaluación de la usabilidad durante la fase de requisitos del desarrollo de software, tomando como insumo la información proporcionada por un Modelo de Casos de Uso (Villegas et al, 2015).

5.1.1 Proceso de incorporación

Desde el punto de vista de HCI, analizar las especificaciones de Casos de Uso por sí solos no tiene mucha utilidad, si lo que se desea es que al final del proceso los desarrolladores comprendan en dónde están los defectos del diseño. En este sentido, es necesario comprender las necesidades e intenciones de los usuarios, lo cual caracteriza el factor humano. Por lo tanto, la principal recomendación es que el equipo de desarrollo debe iniciar un proceso leve de incorporación de HCI a través de la Taxonomía de la Actividad, como se ilustra en la Figura 5.1.

Los prototipos de interfaces y formularios en papel se generan a partir de los modelos mentales (Giraldo, 2010) capturados desde los usuarios representativos. Para este ejemplo, se utiliza el modelo mental capturado para chequear los datos y las descripciones de la actividad especificados en los Casos de Uso, pero es muy importante resaltar que a partir de los modelos mentales también se diseñan los modelos de diálogo y navegación.

Con la información capturada en los modelos mentales, la especificación del Caso de Uso y demás información acerca de los requisitos, un panel de expertos en Ingeniería de Software, en HCI y en el Proceso de Negocio evalúa el Caso de Uso y hace una valoración conceptual con respecto a la especificación del Caso de Uso (inclusión de HCI). La información suministrada también es la base para seguir el proceso de

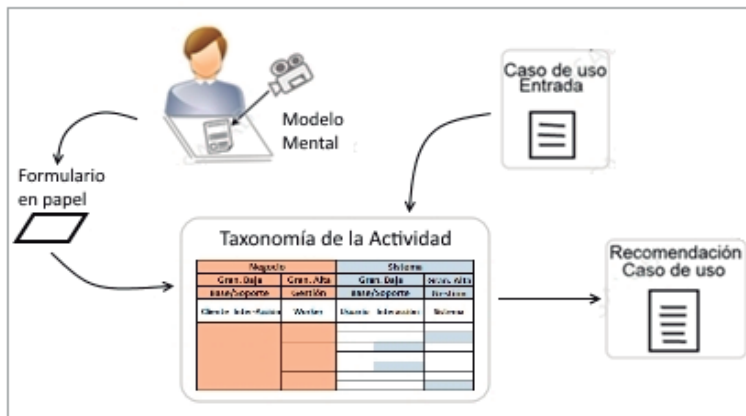


Figura 5.1. Esquema de Incorporación de HCI en un proceso de desarrollo de software a través de la TxA y a partir de los Casos de Uso.

Inicialmente se representa la interacción entre el usuario y el negocio resaltando la importancia del papel. Todo proceso de negocio tiene sus orígenes en el papel. Partiendo de esta idea, los pasos que se describen en las especificaciones de los casos de uso para sistemas interactivos transaccionales, deberían hacerse de la misma manera sin importar el escenario.

Clasificación definido para la Taxonomía de la Actividad. Con la Taxonomía se clasifica la interacción en dos tipos: interacción del negocio (entre personas) e interacción del usuario (persona - máquina). A partir de ésta clasificación y de acuerdo a los modelos mentales capturados de los usuarios representativos, se realizan las recomendaciones y comentarios correspondientes a tener en cuenta en los modelos de diálogo y navegación y en el diseño de las interfaces de usuario según el escenario sobre el cual se vaya a realizar la implementación.

5.1.2 Equipo de expertos

El equipo de expertos debe estar conformado por un grupo de personas que desempeñen los roles asociados con la Taxonomía de la Actividad: *Analista Etnográfico*, *Analista de Requisitos*, *Analista de Sistemas* y *Diseñador de la Interfaz de Usuario*. También intervienen los roles: 1) *Analista del Modelo Mental*: experto en análisis de modelos mentales quien interactúa con usuarios reales del servicio analizado; 2) *Analista de Datos y Forma*: experto en psicología cognitiva que interpreta modelos mentales para extraer información sobre los datos y la forma del modelo mental del usuario; 3) *Diseñador de Prototipos*: experto en datos y forma que elabora prototipado sobre diversos medios de representación: formularios en papel, formularios Web, otros tipos de interfaces; 4) *Analista de Bases de Datos*: experto en bases de datos para proveer feedback acerca de los datos del dominio.

5.1.3 Artefactos

Los artefactos utilizados en el proceso de incorporación se clasifican en insumos y artefactos resultantes.

- **Insumos:** Modelo de Casos de Uso, Modelos Mentales, Reglas de Negocio, Estructura Taxonómica de la Actividad. Para el modelo de casos de uso, se toman como referencia las especificaciones que tengan definidos los Diagramas de Casos de Uso y las Plantillas que especifiquen por lo menos la descripción del

caso de uso, requisitos asociados, descripción de los actores, precondiciones, post-condiciones, flujo básico, y flujos alternos.

- **Artefactos Resultantes:** Documento que contiene la valoración conceptual, comentarios y recomendaciones útiles en el diseño de interacción del sistema.

Generalmente la evaluación de la usabilidad de un sistema se realiza a partir de artefactos como los prototipos de diseño, como se mencionó en la Sección 4.2. Nuestras experiencias han demostrado que la usabilidad no solamente se puede promover desde artefactos que tengan que ver con la IU, puede ser con otros tipos de artefactos principalmente desde las tareas. Esta propuesta está dirigida a realizar un análisis de la funcionalidad del sistema enfocándose en identificar carencias en la especificación de aspectos interactivos que tienen un impacto directo en el diseño de los modelos de interacción, en el diseño de las interfaces de usuario y por tanto en el nivel de usabilidad del producto.

5.1.4 Caso de Estudio

En esta sección se presenta un Caso de Estudio mediante el cual se pretende evaluar la usabilidad de un sistema interactivo a partir de una Especificación de Casos de Uso. Se toma como referencia uno de los Casos de Uso que especifican el funcionamiento del Sistema de Recaudos en una entidad Bancaria a través de un Cajero Electrónico Multifuncional (Figura 5.2). El Caso de Uso seleccionado es "Realizar Recaudo", que describe la funcionalidad del sistema que permite a un usuario pagar una obligación financiera por medio de un Cajero Electrónico utilizando el Teclado como medio de ingreso de los datos. Es importante resaltar que el Caso de Estudio es tomado de un caso real. Por cuestiones de confidencialidad no se revelan los datos de la empresa desarrolladora de software ni de la entidad bancaria.

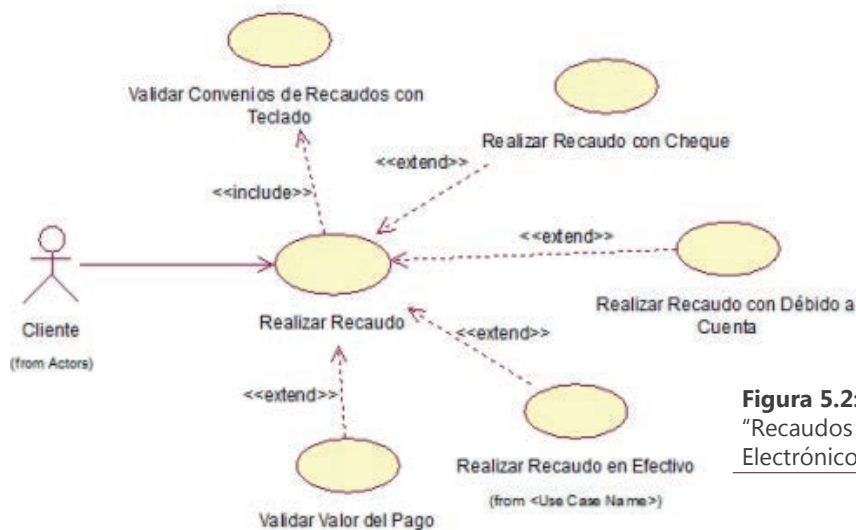


Figura 5.2: Diagrama de Casos de Uso "Recaudos a través de Cajero Electrónico Multifuncional".

Además del Diagrama de Casos de Uso, se toma como referencia la descripción del Flujo Primario del Caso de Uso (Tabla 5.1).



Tabla 5.1. Descripción del flujo primario caso de uso “Realizar Recaudo”.

Flujo Primario “Realizar Recaudo”
1. El Cajero muestra la opción “Pagos a Terceros”
2. El Usuario selecciona la opción “Pagos a Terceros”.
3. El Cajero presenta las opciones para el ingreso de los datos: “Tarjeta Empresarial”, “Teclado” y “Código de Barras”.
4. El Usuario selecciona la opción “Teclado”.
5. El Cajero solicita al usuario ingresar el número de identificación por medio de un mensaje en pantalla: “ <i>Por favor ingresa tú número de identificación personal(C.C., PASAPORTE, TI)</i> ”.
6. El Usuario ingresa número de identificación. (Debe cumplir con Regla de Negocio 1 (RN1)).
7. El Cajero solicita al Usuario que seleccione alguna de las siguientes opciones: “Número de Convenio”, “Número de Cuenta de Ahorros” ó “Número de Cuenta Corriente”. (Debe cumplir con Regla de Negocio 2 (RN2)).
8. El Usuario selecciona la opción “Número de Convenio”. (Remitirse al Flujo Alterno 1 (FA1)).
9. El Cajero presenta en pantalla el campo de texto para digitar el número de convenio y las opciones “Borrar” y “Aceptar”.
10.El Usuario digita el número de convenio. (Debe cumplir con RN3).
11.El Usuario selecciona la opción “Aceptar”. (Remitirse al FA2).
12.El Cajero solicita validar el convenio al Sistema de Recaudos. (Caso de Uso CU-2: Validar Convenios de Recaudos con Teclado) (Debe cumplir con RN4).
13.El Cajero recibe la información generada por el Sistema de Recaudos acerca del convenio. (Debe cumplir con la RN5) (Remitirse al FA3).
14.El Cajero identifica que el estado de la cuenta recaudadora permite continuar con el recaudo. (Remitirse al FA4).
15.El Cajero identifica que el convenio tiene una sólo referencia. (Remitirse al Ver FA5)
16.El Cajero solicita al Usuario que digite el número de la referencia. (Debe cumplir con la RN6) y muestra las opciones “Borrar” y “Aceptar”.
17.El Usuario digita el número de referencia. (Debe cumplir con la RN7)
18.El Usuario selecciona “Aceptar”. (Remitirse al FA6)
19.El Cajero identifica que el número de convenio no existe en las Bases de Datos. (Remitirse al FA7)
20.El Cajero solicita al Usuario que ingrese el valor a pagar y muestra las opciones “Aceptar” y “Borrar”. (Cumplir con las RN7 y RN8).
21.El Usuario digita el valor a pagar.
22.El Usuario selecciona la opción “Aceptar”. (Remitirse al FA8).
23.El Cajero solicita validar el pago al Sistema de Recaudos. (Caso de Uso CU-3: “Validar Valor del Pago”).
24.El Cajero recibe respuesta exitosa del Sistema de Recaudos acerca de la validación del valor. (Remitirse al FA9).
25.El Cajero muestra en pantalla la información referente al recaudo (Debe cumplir con la RN9) y las opciones “Aceptar” y “Cancelar”.
26.El Usuario selecciona la opción “Aceptar”. (Remitirse al FA10)
27.El Cajero muestra en pantalla las opciones para las formas de pago “Débito a Cuenta”, “Cheque” y “Efectivo”.
28.El Flujo continúa de acuerdo a los puntos de extensión {CU-4: “Realizar Recaudo con Débito a Cuenta”}, {CU-5: “Realizar Recaudo con Cheque”}, {CU-6: “Realizar Recaudo en Efectivo”}.

De acuerdo con la especificación del Flujo Primario para el Caso de Uso “Realizar Recaudo”, se presenta a continuación la aplicación del Esquema de Incorporación de HCI a través de la Taxonomía de la Actividad con el fin de obtener una valoración del Caso de Uso con respecto a HCI, específicamente el nivel de Usabilidad.

5.1.4.1 Captura del Modelo Mental del Usuario

Esta actividad se fundamenta en el hecho de que todo proceso de negocio tiene sus orígenes en el papel. Se destaca la interacción entre el usuario y el negocio. Así, los pasos a seguir para realizar el pago de una obligación financiera, según la especificación del Caso de Uso suministrado, debería realizarse de la misma manera sin importar el escenario, tomando como escenarios posibles: (1) Realizar el pago en el banco utilizando formularios en papel, (2) Realizar el pago a través de un cajero multifuncional interactuando con pantallas, o (3) Realizar el pago a través de un aplicativo web.

La Figura 5.3 presenta el modelo mental asociado a la transacción de pagos a terceros. Dicho modelo mental fue capturado luego de realizar un taller de discusión dirigida (*Focus Group*) en el que participaron usuarios con experiencia realizando transacciones bancarias. Los detalles del proceso utilizado para generar el modelo mental no corresponden al alcance de esta sección.

Recauda Banco

CUIDAD: [] DÍA: [] MES: [] AÑO: []

INFORMACIÓN DEL PAGADOR

NOMBRE DEL PAGADOR: []

REFERENCIA(S): [] TELEFONO: []

INFORMACIÓN DEL BENEFICIARIO

CÓDIGO DE CONVENIO: []

TITULAR DE LA CUENTA O NOMBRE DEL CONVENIO: []

CUENTA CORRIENT: []

CUENTA AHORROS: []

NÚMERO DE CUENTA: []

OBLIGACIONES

OBLIGACIONES	
CONCEPTO	VALOR

FORMA DE PAGO

Total Efectivo: \$ [] + Total Cheques: Cantidad [] \$ [] = Total: \$ []

CHEQUES

CHEQUES		
BANCO	CHEQUE No	VALOR \$

Figura 5.3. Formulario resultante a partir de la captura del Modelo Mental de usuarios representativos.

La información de cómo un usuario quiere manipular la información del modelo mental a medida que avanza en el proceso de negocio es capturada y especificada en los modelos de navegación y de diálogo. Para este caso nos interesa utilizar el formulario resultante para guiar a los expertos en el proceso de verificación de la especificación del Caso de Uso.

5.1.4.2 Evaluación del Caso de Uso utilizando la TxA

El Proceso de Clasificación de la Actividad permite identificar los niveles de abstracción y granularidad utilizados en el Caso de Uso que se evaluará. Se clasifica la interacción en dos tipos: interacción del negocio (entre personas) e interacción del usuario



(persona - máquina), teniendo claridad en los tipos de interacción que se presentan en el caso de uso, se realizan las recomendaciones correspondientes al feedback que debe generarse por parte del sistema hacia el usuario.

En la especificación del Caso de Uso "*Realizar Recaudo*" (Tabla 5.1), se observa que no se diferencia la interacción del negocio de la interacción del sistema. Por esta razón se realiza un análisis adicional de la Actividad (Tabla 5.2. Nivel de Negocio), para que sea posible presentar la información del Caso de Uso "*Realizar Recaudo*" clasificada de acuerdo a la Estructura de la TxA, para la columna Función (sólo Actividad) (Tabla 5.2).

Se diferencian por colores los niveles de abstracción Negocio y Sistema. Las celdas marcadas con amarillo, indican los pasos del Caso de Uso donde se hacen recomendaciones en el reporte de incidencias. Para mayor claridad sólo se ha hecho referencia a los pasos especificados en el flujo básico. Del mismo modo en que se hace referencia, por ejemplo a una Regla de Negocio (RN) o a un Flujo Alterno (FA) dentro de la especificación del Caso de Uso (TablaLa nomenclatura es: **FB** (Feedback), **AW** (Awareness), **IP** (Patrones de Interacción), **COM** (Comentarios). Con estas marcas, los comentarios realizados sobre los casos de uso, generados por el equipo de pruebas, tienen suficiente contexto como para entender a qué se refiere dicho comentario.

5.1.4.3 Reporte con Comentarios y Recomendaciones

La estructura del reporte inicia con un listado de observaciones generales al Caso de Uso, seguido por el listado de comentarios y recomendaciones relacionados con cada uno de los pasos especificados en el Flujo Primario, a fin de tener en cuenta modelos de diálogo y navegación en el diseño de las interfaces de usuario, según el escenario sobre el cual se vaya a realizar la implementación. Para el Caso de Estudio analizado, en esta sección las tablas 5.3 y 5.4 presentan el listado de observaciones y el listado de comentarios y recomendaciones, respectivamente.

5.1.4.4 Resultados

A partir de los resultados obtenidos en el Caso de Estudio se puede observar que en la Especificación del Caso de Uso no se diferencia la especificación de las tareas atemporales o de negocio. La definición de este tipo de tareas y su interpretación a través del escenario sobre el cual va a ser implementado el sistema es de vital importancia porque de este modo el usuario puede diferenciar información sobre feedback de negocio y de sistema, awareness de negocio y de sistema, etc. Adicionalmente, el hecho de identificar carencias en la definición de Patrones de Interacción, Tareas de Feedback, Tareas de Awareness, a partir de la especificación de la funcionalidad de un sistema, permite incorporar aspectos del diseño de la interacción, del diseño de interfaces de usuario, desde el inicio del proceso de desarrollo de los productos software. Esta incorporación aporta positivamente en los niveles de usabilidad del sistema. Tenemos que por ejemplo para este trabajo se pasó de tener cero (0) Tareas de Feedback, cero (0) Tareas de Awareness, cero (0) Necesidades de Patrones de Interacción, a tener cinco (5) Tareas de Feedback, cinco (5) Tareas de Awareness, cuatro (4) Necesidades de Patrones de Interacción, con las cuales se debería complementar la especificación del Caso de Uso "*Realizar Recaudo*" para mejorar HCI.

Tabla 5.2 (a). Información del Caso de Uso “Realizar Recaudo” ubicada sobre la Estructura Taxonómica de la Actividad.

Negocio			Sistema				
Base/Soporte		Gestión	Base/Soporte		Gestión		
Cliente	Inter-Acción	Worker	Usuario	Interacción	Sistema Cajero	Sistema Recaudos	
Elegir beneficiario del pago (terceros)			Elegir beneficiario del pago (terceros)				
	El Cliente solicita al Cajero un formulario para pagos a terceros.			Paso 1			
	El Worker entrega al Cliente un formulario para pagos a terceros.			Paso 2			
Organizar la información referente al beneficiario del pago.			Organizar la información referente al beneficiario del pago.				
Verificar que la información básica del formulario para pagos diligenciada sea correcta	El Cliente diligencia información básica del formulario para pagos a terceros.	Verificar que la información diligenciada por el Cliente corresponda con el formato permitido en los campos del formulario.		Paso 3 (COM1)			
				Paso 4			
				Paso 5 (IP1)			
				Paso 6 (FB1)			
				Paso 7 (FB2)			
			Identificar el tipo de cuenta o convenio del beneficiario o recaudador del pago				
Organizar información referente al convenio.			Organizar información referente al convenio.				
Verificar que la información del convenio diligenciada sea correcta	El Cliente diligencia información del Convenio.			Paso 8			
				Paso 9 (COM2)			
				Paso 10 (FB2)			
				Paso 11			
		Verificar la información del número de convenio.				Paso 12 (AW1)	
							Validar el convenio
						Paso 13 (AW2)	
				Paso 14 (AW1)			



● **Tabla 5.2 (b).** Información del Caso de Uso “Realizar Recaudo” ubicada sobre la Estructura Taxonómica de la Actividad.

Negocio		Sistema					
Base/Soporte		Gestión	Base/Soporte		Gestión		
Cliente	Inter-Acción	Worker	Usuario	Interacción	Sistema Cajero	Sistema Recaudos	
Verificar que la información del convenio diligenciada sea correcta	El Cliente diligencia información del Convenio.	Verificar la información de la referencia del convenio.			Paso 15 (AW1)		
				Paso 16 (IP2)			
				Paso 17 (FB4)			
			Verificar que la referencia del convenio diligenciado sea el correcto				
				Paso 18			
					Paso 19(AW1)		
		Verificar el valor de pago del convenio.		Paso 20 (IP3) (COM2)			
				Paso 21 (FB5)			
			Verificar que el valor a pagar ingresado sea el correcto				
				Paso 22			
						Paso 23 (AW4)	
							Validar el valor a pagar
						Paso 24 (AW5)	
					Paso 25(IP4)(COM3)		
	Verificar que la información sea correcta						
			Paso 26				
			Paso 27				
			El Usuario analiza la forma de pago de acuerdo a sus necesidades.				

Los resultados también nos permiten concluir que desde el punto de vista de HCI, específicamente de la evaluación de la usabilidad, no resulta muy útil realizar un Análisis de Casos de Uso de una manera aislada de los usuarios. Por esta razón se tiene en cuenta la información capturada en los modelos mentales para chequear los aspectos interactivos que tienen lugar en los flujos de las actividades analizadas. Por

otro lado, si nos referimos a las especificaciones de la labor capturadas por un ingeniero de desarrollo a través de plantillas centradas en flujos y desde su propia perspectiva, se concluye que dichas especificaciones no incluyen los aspectos de HCI necesarios para el desarrollo de aplicaciones interactivas. Es decir, las plantillas actuales que se utilizan para especificar los Casos de Uso, no resultan estar lo suficientemente estructuradas para capturar información de la estructura y función de la interfaz de usuario. En este sentido, es recomendable definir una nueva plantilla que permita capturar información complementaria a la especificación de los Casos de Uso, con respecto a incidencias relacionadas con la usabilidad.

Tabla 5.3. Listado de observaciones generales al Caso de Uso “Realizar Recaudo”.

Listado de Observaciones Generales al Caso de Uso “Realizar Recaudo”	
Observación	Detalle
Desde el punto de vista de la incorporación de HCI se recomienda tener una plantilla enriquecida para casos de uso	<p>Comprender las carencias y excesos de usabilidad (HCI) en especificaciones de casos de uso implica proveer contexto que proviene de las interacciones entre humanos y que posteriormente dirigen las interacciones humano computador.</p> <p>A modo de ejemplo, se puede observar cómo en el paso 6 del CU (Tabla 16) (<i>El actor ingresa el número de identificación (Ver RNI)</i>) se hace referencia a la regla de negocio RNI y uno puede encontrarla en el mismo documento, aun cuando las reglas no son especificaciones de la labor (función, el cómo) sino de la motivación (el por qué). De esta misma forma el equipo de pruebas podría incluir observaciones o sugerencias, cuando en un paso lo requiera, a elementos que se refieran a plantillas, patrones de interacción, contenedores concretos de la interfaz de usuario (feedback, awareness), etc. Esto sustenta la necesidad de una nueva plantilla que incorpore elementos de HCI para la especificación de casos de uso. Adicionalmente, los comentarios del equipo de pruebas realizados sobre los casos de uso van a tener el suficiente contexto como para entender a qué se refiere dicho comentario.</p> <p>En este sentido, algunas observaciones en el paso 6 del CU (<i>El actor ingresa el número de identificación (Ver RNI)</i>) podrían ser:</p> <ul style="list-style-type: none"> • Utilizar el patrón de interacción de ingreso de datos. • Proveer feedback de negocio utilizando la plantilla PL001.
Las especificaciones se han realizado desde el punto de vista de la compañía y no desde la perspectiva del usuario.	<p>Se recomienda que la semántica de los servicios y sus especificaciones se hagan enfocadas en proveer un resultado de valor al usuario.</p> <p>Cuando un usuario decide utilizar un servicio de negocio en una organización es porque dicha organización le provee un resultado de valor que está en coherencia con las necesidades e intenciones del usuario (semántica). En un proceso de recaudo de terceros es una tercera compañía la que desea hacer unos recaudos por intermediación del banco. Es así como, el banco ofrece un servicio a dos clientes: la tercera compañía y el usuario de la tercera compañía. En este sentido, el usuario ni siquiera tiene que ser un cliente del banco y por tanto no tiene por qué conocer la mecánica del banco. La tercera compañía hace recaudo mientras que el usuario paga sus obligaciones. Un cliente que paga sus obligaciones por intermedio de un banco ya sea apoyado por un ATM o por un cajero de oficina no entiende por qué está pagando recaudos.</p>

Tabla 5.4. Reporte de Comentarios y Recomendaciones al Caso de Uso "Realizar Recaudo".

Comentarios y Recomendaciones al Caso de Uso	
Atributo	Recomendación
Recomendaciones sobre Feedback	FB1: Se debe indicar el paso a seguir por parte del usuario o sistema luego de que éste ingresa el número de identificación., es decir, cómo sigue la interacción (presionar un botón, aceptar en una opción de pantalla..).
	FB2: Según las reglas del negocio se puede corregir dos veces el número de convenio. Se debe indicar esto al usuario.
	FB3: Según las reglas del negocio se puede corregir dos veces el número de convenio, se debe indicar esto al usuario.
	FB4: Feedback del sistema: Se debe informar al usuario que si corrige dos veces se cancela la transacción.
	FB5: Se le debe informar al usuario el número de veces que se permite corregir el valor digitado por el usuario.
Recomendaciones sobre Awareness	AW1: Se debe indicar al usuario que se está validando el número de convenio.
	AW2: Se debe indicar al usuario que se está validando el número de convenio. Esta sección, donde se hace la validación del convenio, es necesario prestar atención a la interacción del sistema con el usuario porque se pueden generar diferentes excepciones que acarrear mensajes de feedback al usuario.
	AW3: Se debe indicar al usuario que se están validando las referencias.
	AW4: Se debe indicar al usuario que se está validando el valor a pagar, debido a que las validaciones que hace el sistema pueden tomar algunos segundos.
	AW5: Se debe indicar al usuario que se está validando el valor a pagar. Es necesario prestar atención a la interacción del sistema con el usuario porque se pueden generar diferentes excepciones que acarrear mensajes de feedback al usuario.
Patrones de Interacción	IP1: Para actividades en las que se requiere ingresar información, se sugiere definir un conjunto de "Patrones de Interacción". Para este caso, por ejemplo, un patrón para "Ingresar Información".
	IP2: Se sugiere definir un patrón de ingreso de información donde se indique el texto de encabezado (debe salir el nombre de la referencia que tenga recaudos, por ejemplo: cupón, NIT, referencia, etc) para esta pantalla.
	IP3: Se sugiere definir un patrón de ingreso de información.
	IP4: Se sugiere definir un patrón de presentación de la información.
Comentarios	COM1: Se debe mejorar la redacción de este paso, indicando el texto, la semántica de la operación. Se sugiere utilizar la siguiente estructura semántica: " <i>Selecciona el medio de ingreso de datos: Teclado, Código de barras, Tarjeta empresarial</i> "
	COM2: Se sugiere evaluar, mediante pruebas con usuarios, si la opción debe ser llamada "Borrar", o más bien "Corregir". De acuerdo a la experiencia de los expertos en el uso de Cajeros Automáticos, la opción "Borrar" existe en el teclado numérico de los Cajeros. Se sugiere evaluar, puede ser mediante pruebas con usuarios, si la opción "Borrar" se debe incorporar a las opciones que se muestran en pantalla o si los usuarios ya se encuentran familiarizados con la opción por teclado.
	COM3: Según el documento de requisitos primero se muestra la información y posteriormente se proceden a realizar las respectivas validaciones por parte del sistema. Se recomienda validar primero antes de mostrar la información al usuario.

5.2 EVALUACIÓN DE LENGUAJES DE MODELADO

El interés principal de esta sección es introducir cómo es posible utilizar la TxA para evaluar un conjunto de elementos notacionales que se utilizan actualmente para modelar la actividad (labor) en los sistemas interactivos. Se presentan dos dinámicas de evaluación. La primera, enfocada en clasificar elementos notacionales de acuerdo al proceso de clasificación definido para la TxA. La segunda, enfocada a mostrar los resultados obtenidos en el proceso de evaluar un conjunto de elementos de modelado para la especificación de Sistemas Interactivos, a partir de los clasificadores definidos en la Taxonomía de la Actividad (TxA).

5.2.1 Clasificación de Elementos de Modelado a partir de la TxA

En la sección 2.3 se realizó un análisis preliminar de un conjunto de propuestas existentes para el desarrollo de sistemas interactivos, donde se clasificaron sus elementos notacionales. Posteriormente, en el capítulo 3 se realizó un análisis de propuestas extendido debido a que se incrementó el número de clasificadores que conforman la Taxonomía. No solamente se consideró la clasificación de la Actividad (Cómo), sino que también se agregaron otras vistas utilizadas para el modelado de sistemas interactivos, esto es: el modelado de Datos (Qué), Personas (Quién), Ubicación (Dónde), Tiempo (Cuándo) y Motivación (Por qué). Lo anterior, debido a que se debe entender el modelado de sistemas interactivos de manera integral. El modelado de la actividad debe relacionarse con los demás modelos que representan el sistema interactivo completo, Modelos de la Organización, de los Datos, de los Lugares, etc.

El propósito de esta sección es presentar la clasificación de los elementos de modelado, propuestos por el Lenguaje UML y la propuesta CIAM (Molina, 2006), de acuerdo con el proceso de clasificación definido para la TxA en la sección 3.3. La clasificación tiene como fin poder utilizar la taxonomía como marco de evaluación de propuestas que soportan el modelado de sistemas interactivos (y software en general), específicamente el modelado de la actividad. Lo que se clasifica es la notación y no la metodología porque la notación es la que describe la información del producto, en este caso el sistema interactivo.

Siguiendo los pasos de Clasificación para la Taxonomía de la Actividad (Tabla 3.7), las propuestas metodológicas y los elementos de modelado que proveen las notaciones UML y CIAM se clasifican como sigue:

1. Relevancia: OpenUp y CIAM califican para ser clasificadas con la taxonomía.
2. Caracterización Básica:
 - a. OpenUp: se enfoca en el aspecto de la funcionalidad.
 - b. CIAM: se enfoca en el aspecto de la colaboración.



3. Niveles de abstracción:

- a. UML: provee elementos de modelado para los niveles de abstracción Negocio y Sistema.
- b. CIAN: provee elementos de modelado sólo a nivel de Sistema.

4. Tipo de Actividad:

- a. UML: usa el mismo elemento de modelado, sin hacer una distinción sintáctica para soportar de manera separada las categorías base, soporte y gestión.
- b. CIAN: usa los mismos elementos de modelado para soportar las categorías base y soporte. Hace una distinción sintáctica para representar la categoría gestión.

5. Especificación a nivel de negocio:

- a. UML: usa el mismo elemento de modelado, sin hacer una distinción sintáctica para soportar las intenciones del actor de negocio y las intenciones del worker. Provee el elemento de modelado Asociación para representar las interacciones.
- b. CIAN: No provee elementos de modelado en el nivel de abstracción de Negocio.

6. Nivel de granularidad a nivel de negocio:

- a. UML: Para diferenciar el nivel de granularidad bajo en el que se especifican las actividades a nivel de negocio provee los elementos de modelado Caso de Uso de Negocio y Diagrama de Actividad.

7. Roles a nivel de negocio:

- a. UML: Provee los elementos de modelado Actor de Negocio y Worker para diferenciar los roles que intervienen en las actividades a nivel de negocio.

8. Metas-Objetivos a nivel de negocio:

- a. UML: Provee el elemento de modelado Meta para clasificar los objetivos a los cuales contribuye la ejecución de las actividades a nivel de negocio.

9. Entidades a nivel de negocio:

- a. UML: Provee el elemento Entidad de Negocio para clasificar las entidades – datos relacionados con la ejecución de las actividades a nivel de negocio.

10 y 11. Lugares y tiempos a nivel de negocio:

- a. UML: No provee elementos de modelado para clasificar los lugares ni los tiempos en los cuales toma lugar la actividad a nivel de negocio. Aunque en



el diagrama de actividad sí proporciona elementos para representar sincronización entre actividades.

12. Especificación a nivel de sistema::

- a. UML: usa el mismo elemento de modelado, sin hacer una distinción sintáctica para soportar las intenciones del actor de sistema y las acciones del sistema. Provee el elemento de modelado Asociación para representar las interacciones.
- b. CIAN: hace una distinción sintáctica para soportar las intenciones del actor de sistema y las acciones del sistema. Provee elementos de modelado (tipos de tareas) para representar las interacciones.

13. Nivel de granularidad a nivel de sistema:

- a. UML: Para diferenciar el nivel de granularidad en el que se especifican las actividades a nivel de sistema provee los elementos de modelado Caso de Uso de Sistema Compuesto, Caso de Uso de Sistema y Acción de Sistema.
- b. CIAN: Para diferenciar el nivel de granularidad en el que se especifican las actividades a nivel de sistema provee los elementos de modelado Tarea Abstracta CIAN, Tarea Abstracta CTT, Tarea Cooperativa, Tarea Colaborativa, Tarea de Usuario CTT, Tarea Interactiva CTT, Tarea de Aplicación CTT.

14. Roles a nivel de sistema:

- a. UML: Provee el elemento de modelado Actor de Sistema para diferenciar los roles que intervienen en las actividades a nivel de sistema.
- b. CIAN: Provee los elementos de modelado Actor, Rol, Agente Software, Grupo, Equipo de Trabajo, para diferenciar los roles que intervienen en las actividades a nivel de sistema

15. Metas-Objetivos a nivel de sistema:

- a. UML: No provee elementos de modelado para clasificar los objetivos a los cuales contribuye la ejecución de las actividades a nivel de sistema.
- b. CIAN: Provee elementos de modelado para clasificar elementos relacionados con reglas a nivel de sistema, Área de Visualización Colaborativa, Área de Visualización Individual, Segmento de Edición Exclusiva.

16. Entidades a nivel de sistema:

- a. UML: Provee el elemento Clase para clasificar las entidades relacionadas con la ejecución de las actividades a nivel de sistema.
- b. CIAN: Provee el elemento Objeto para clasificar las entidades relacionadas con la ejecución de las actividades a nivel de sistema.



17. Lugares a nivel de sistema:

- a. UML y CIAN: No proveen elementos de modelado para clasificar los lugares en los cuales tiene lugar la actividad a nivel de sistema.

18. Tiempos a nivel de sistema:

- a. UML: No provee elementos de modelado para clasificar los tiempos en los cuales toma lugar la actividad a nivel de sistema.
- b. CIAN: Provee el elemento Restricción Temporal Cuantitativa para clasificar los tiempos en los cuales toma lugar la actividad a nivel de sistema. Provee también los operadores temporales para relacionar las tareas.

Las figuras 5.4 y 5.5 muestran la clasificación realizada según la taxonomía de la actividad, para los elementos notacionales de UML y CIAN. Esta representación permite una mejor visualización de los aportes y carencias de cada una. Es decir, sería posible medir un atributo como la cobertura. Según Giraldo (Giraldo, 2010), la cobertura está determinada por el número de celdas que el elemento sometido a evaluación, es capaz de describir. La clasificación realizada permite entonces visualizar en qué nivel las notaciones evaluadas cubren las celdas definidas en la estructura taxonómica de la actividad, es decir, cuáles clasificadores soportan para el modelado de sistemas interactivos.

19. Conclusión:

Se observa que el UML provee elementos de modelado de la actividad en los que se separa muy bien los niveles de abstracción de Negocio y de Sistema, aunque sólo se centra en modelar el aspecto de la Funcionalidad. Para el lenguaje CIAN se observa que sólo las categorías del nivel del sistema están soportadas por elementos de modelado. Esto es debido a que las tareas en CIAN están asociadas directamente a roles, que son usuarios del sistema informático, y porque CIAN no contempla el modelado de los usuarios del negocio, de manera equivalente a los actores de negocio.

En la clasificación del modelado de la actividad para UML, cada tipo de actividad está soportada por, al menos, un elemento de modelado, sin hacer una distinción sintáctica para soportar de manera separada las categorías base, soporte, y gestión. Aun así, es posible clasificar las actividades de un sistema informático en dichas categorías. Esta falta de elementos de modelado de UML genera un bajo nivel de expresividad y semántica para modelar otros aspectos como la interacción con el usuario, la colaboración, etc.

La clasificación en CIAN da como resultado que el nivel alto de granularidad está representado por las tareas CTT y las herramientas de soporte independientes de la actividad, y en el nivel bajo de granularidad está representado por las tareas de inter-acción.

Estructura Taxonomía de la Actividad: Nivel Negocio UML y CIAN										
Gran/Elem	Base - Soporte			Gestión		Dato	Rol	Meta	Lugar	Tiempo
	C	I-A	W							
G. Baja	UML CIAN	UML CIAN	UML CIAN	UML CIAN	UML CIAN	UML CIAN	UML CIAN	UML CIAN	UML CIAN	UML CIAN
G. Alta										

Figura 5.4. Clasificación de los elementos notacionales de UML y CIAN según la estructura de la Taxonomía de la Actividad, Nivel de abstracción Negocio.

Estructura Taxonomía de la Actividad: Nivel Sistema UML y CIAN									
Gran/Elem	Base - Soporte		Gestión		Dato	Rol	Meta	Lugar	Tiempo
	U	I	S	W					
G. Baja	UML CIAN	UML CIAN	UML CIAN	UML CIAN	UML CIAN	UML CIAN	UML CIAN	UML CIAN	UML CIAN
G. Alta									

Figura 5.5. Clasificación de los elementos notacionales de UML y CIAN según la estructura de la Taxonomía de la Actividad, Nivel de abstracción Sistema.

A partir de la evaluación de propuestas es posible analizar posibilidades de integración, identificación de puntos de intercambio con el fin de aprovechar al máximo sus características.

5.2.2 Selección de elementos de modelado para la especificación de sistemas interactivos (Villegas et al, 2016b)

El interés principal de esta sección es analizar y evaluar un conjunto de elementos notacionales que se utilizan actualmente para modelar la actividad (labor) en los sistemas interactivos, a partir de los clasificadores definidos en la Taxonomía de la Actividad (TxA). Se pretende encontrar una correspondencia entre los clasificadores definidos en la Taxonomía y los elementos de modelado seleccionados. La evaluación está orientada también a identificar los elementos de modelado más utilizados por la comunidad de expertos en Ingeniería de Software (IS) y en Interacción Humano Computador (HCI). No se intenta medir el nivel de separación que existe entre dichas comunidades, porque esto ya es algo obvio. Más bien lo que se pretende es, desde un elemento que es común para todos que es la Actividad, plantear la necesidad de acercar el HCI a la IS y viceversa, porque a fin de cuentas, ambas se enfocan en producir software.

5.2.2.1 Metodología de trabajo

El proceso de evaluación realizado en este trabajo inicia con la planeación de la evaluación, seguida de la preparación, de la captura de información y por último, del análisis de resultados (Figura 5.6).

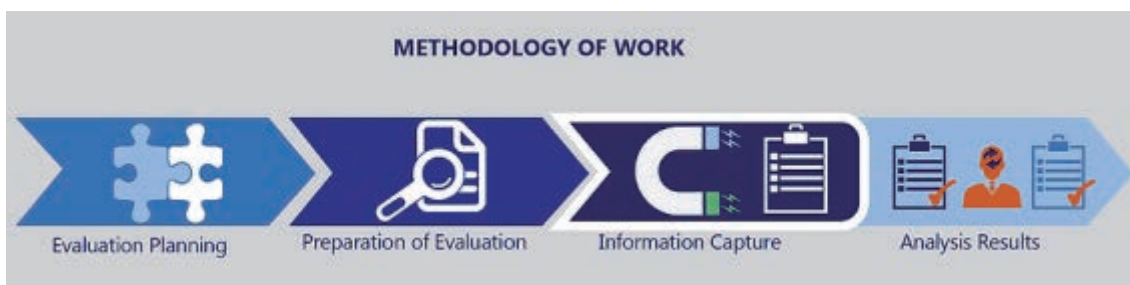


Figura 5.6. Metodología de trabajo de dirige el proceso de evaluación.

5.2.2.1.1 Planeación de la evaluación

El diseño de la evaluación presentada en esta sección se conduce de acuerdo con los siguientes requisitos:

- ⦿ Los instrumentos de evaluación necesariamente deben contener un documento de contexto, donde se describa la estructura taxonómica de la TxA y se describan detalladamente los clasificadores con un caso de estudio. Una mejor contextualización permite capturar con mayor

precisión los modelos mentales de los participantes y analizar su potencial para comprender lo relacionado con el modelado de los sistemas interactivos. Adicionalmente, permite que los encuestados tengan el mismo nivel de conocimiento sobre los elementos notacionales a evaluar.

- ⦿ Los perfiles de los encuestados deben estar clasificados teniendo en cuenta: Nivel de Experiencia, Nivel de Formación y Nivel de Conocimiento, tanto en Ingeniería de Software como en HCI. Esto está dirigido a tener resultados concluyentes porque en este tipo de estudios influye demasiado la subjetividad de los participantes.

Luego de definir los requisitos anteriores, se procede a decidir qué artefactos serán necesarios para llevar a cabo la evaluación.

5.2.2.1.2 Preparación de la evaluación

La preparación de la evaluación consiste en definir los siguientes artefactos: primero, un instrumento que permita clasificar los perfiles de los encuestados; segundo, un documento de contexto que permita de cierta forma, nivelar el conocimiento que tienen los encuestados acerca de los clasificadores que define la TxA para el modelado de la labor, la actividad en los sistemas interactivos; y por último, el documento que contiene las preguntas orientadas a la selección de los elementos de modelado (Figura 5.7).

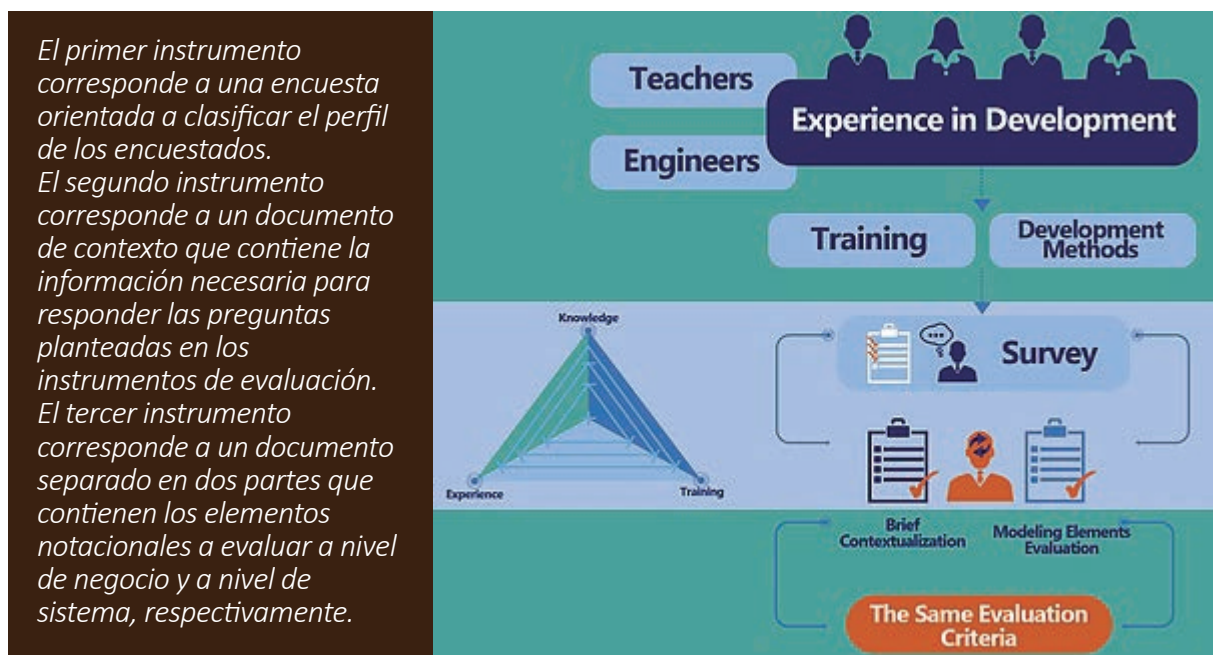


Figura 5.7. Esquema de preparación de la evaluación.



El primer instrumento corresponde a una encuesta orientada a clasificar el perfil de los encuestados. Captura información sobre el nivel de experiencia, formación y conocimiento de los participantes, en Lenguajes de Modelado tanto en IS como en HCI. El interés principal de este instrumento era seleccionar los distintos perfiles, mediante las siguientes variables:

- ⦿ Profesionales en informática o afín, expertos en HCI, **con** y **sin** experiencia en Desarrollo de Software.
- ⦿ Profesionales en informática o afín, **NO** expertos en HCI, **con** y **sin** experiencia en Desarrollo de Software.

Y finalmente, tener resultados más concluyentes que en trabajos anteriores sobre el entendimiento que estas personas tienen sobre los elementos de modelado que seleccionen en las encuestas. Este instrumento de evaluación se puede apreciar en detalle en el **Anexo A**.

El segundo instrumento corresponde a un documento de contexto que contiene la información necesaria para responder las preguntas planteadas en los instrumentos de evaluación. Este documento de contexto se encuentra en el **Anexo B**.

El tercer instrumento corresponde a un documento separado en dos partes que contienen los elementos notacionales a evaluar a nivel de negocio y a nivel de sistema, respectivamente. Se separaron debido a la cantidad de clasificadores definidos por la TxA en cada nivel de abstracción. Es importante resaltar que la clasificación de los elementos notacionales, de acuerdo a la estructura taxonómica de la TxA, fue realizada previamente. Los documentos que contienen estas encuestas se pueden consultar en el **Anexo C**.

Los instrumentos de evaluación se diseñaron utilizando la herramienta de formularios de google © principalmente porque la mayoría de los participantes se encuentran dispersos geográficamente y también porque dichas herramientas permiten recolectar y procesar la información al mismo tiempo.

La información que se pretende capturar con los cuestionarios, tiene que ver con variables que miden el nivel de correspondencia entre los clasificadores definidos en la TxA y los elementos de modelado seleccionados, el nivel de cobertura que tiene cada propuesta metodológica analizada con respecto a los clasificadores definidos en la TxA y el número de elementos de modelado más utilizados por la comunidad de expertos en IS y en HCI.

5.2.2.1.3 Captura de la información

Se solicitó la colaboración de 17 profesionales en Ingeniería de Software, Informática y afines con amplia experiencia en desarrollo de software, que laboran en empresas desarrolladoras de software y que también producen aplicaciones software para el sector gubernamental. Algunos de ellos se consideran expertos en el área de HCI y también tienen experiencia en docencia universitaria en las áreas mencionadas.

Se indicó a cada participante la secuencia en la que debían contestar los cuestionarios. Primero, se debía contestar la encuesta sobre Perfil de los Encuestados; segundo, se debía estudiar el Documento de Contexto y; por último, se debía contestar la Encuesta sobre Selección de los Elementos de Modelado.

5.2.2.1.4 Análisis de resultados

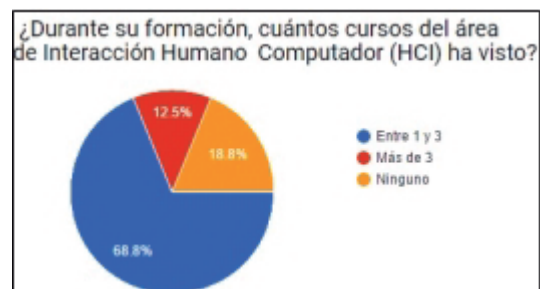
El análisis de los resultados está orientado a concluir cuáles serían los elementos notacionales que potencialmente representan mejor los clasificadores definidos para la TxA y también analizar si el nivel de separación entre las comunidades de IS y HCI permite diseñar mecanismos de incorporación de HCI en los procesos de desarrollo de software y viceversa. De esta forma se podría entender si amerita diseñar un lenguaje para la comunidad académica y para la industria.

- **Resultados Obtenidos Según los Perfiles de los Encuestados:** Los resultados para la definición de los perfiles de las personas encuestadas, permiten rescatar lo siguiente:
 - El mayor nivel de titulación alcanzado hasta ahora por las personas encuestadas es Maestría (Figura 5.8).
 - La mayoría de los encuestados tiene poca formación en HCI (Figura 5.9).
 - Sólo el 36% de los encuestados que se consideran expertos en HCI, han desarrollado productos de mercado, basado en un portafolio aplicando técnicas de HCI. (Figura 5.10). Este porcentaje es un indicador de que existe una separación entre las comunidades de IS y HCI.



Figura 5.8. Nivel de titulación de las personas encuestadas

Figura 5.9. Nivel de formación en HCI de las personas encuestadas.



- Todas las personas que se consideran expertas en Desarrollo de Software (44%), aseguraron que el lenguaje de modelado que utilizaban en sus proyectos es UML. En un porcentaje menor, aseguraron el haber utilizado BPMN y unos cuantos aseguraron el haber utilizado CTT y WISDOM. Esto refleja una gran aceptación de la comunidad de IS por el lenguaje UML, pero también indica que quienes se desempeñan en el campo empresarial tienen gran nivel de desconocimiento acerca de los modelos necesarios en el desarrollo de sistemas interactivos, como los modelos de tareas, modelos de navegación, de diálogo, etc, definidos por la comunidad de expertos en HCI.











Figura 5.10. Nivel de experiencia vs. nivel de aplicación de técnicas de HCI.

- **Resultados Obtenidos en la Selección de Elementos de Modelado para la TxA:** Los resultados obtenidos en la selección de los elementos de modelado y que corresponden con los clasificadores de la TxA se encuentran condensados en las Figuras 5.11 y 5.12.

Cómo					
Tarea que hace el Cliente	Inter-Acción			Tarea que hace el Worker	Datos
	Inter-Acción	Feedback de Negocio	Awareness de Negocio		
<p>(CIAN)</p>	<p>(UML)</p>	Ninguno	Ninguno	<p>(CIAN)</p>	<p>(UML)</p>
Meta		Quién		Ubicación	Tiempo
Objetivo	Regla	Cliente	Worker		
Ninguno	<p>(BPMN)</p>	<p>(UML)</p>	<p>(UML)</p>	<p>(UML)</p>	<p>(CIAN)</p>

Figura 5.11. Elementos seleccionados para los clasificadores en el nivel de negocio.

Lo que salta a simple vista en los resultados obtenidos es que la mayoría de encuestados seleccionaron los elementos de modelado del lenguaje UML, lo cual refleja que el nivel de conocimiento y familiaridad influye bastante a la hora de la selección. Se observa que en los casos en los que no seleccionaron UML, por ejemplo, para el Clasificador <Tarea que hace el Cliente> (Figura 5.11), posiblemente fue porque el lenguaje no provee elementos de modelado para representar dicho clasificador.

Tarea que hace el Usuario	Cómo				Datos
	Interacción			Tarea que hace el Sistema	
	Interacción	Feedback de Sistema	Awareness de Sistema		
 El Cajero decide qué tipo de transacción realizar (BPMN)  Decidir cuál transacción realizar (CIAN)	 (UML)  (HAMSTERS)	 (CTT)	 (HAMSTERS)	 (UML)	 (UML)


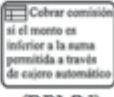




Meta		Quién		Ubicación	Tiempo
Objetivo	Regla	Usuario	Sistema		
 Transacción comprometida en Sistema Local y Sistema Central (UML)	 (BPMN)	 (UML)	 (Workflow)	 (UML)	 (CIAN)

Figura 5.12. Elementos seleccionados para los clasificadores en el nivel de sistema.

Se nota también que la mayoría de los expertos encuestados no diferenciaron los elementos de modelado para representar los clasificadores <Cliente> y <Worker> (Figura 5.12) a nivel de Negocio y los clasificadores <Regla> y <Tiempo> a nivel de Negocio y Sistema. Adicionalmente, que no se seleccionaron elementos de modelado para representar <Feedback y Awareness> a nivel de Negocio (Figura 5.11).

En algunos casos los elementos seleccionados por el nivel de experiencia, correspondían a los de UML, pero según la expresividad y lo que daba a entender a los encuestados, resultaban ser los de otras propuestas, como el caso del clasificador <Interacción> (Figura 5.12).

Un análisis más detallado de los resultados obtenidos permite concluir que para la especificación de la labor, en primer lugar, no se diferencian los elementos notacionales en los niveles de abstracción Negocio y Sistema. En segundo lugar, que las notaciones existentes no proveen elementos de modelado para especificar la interacción a nivel de negocio, como lo es el *feedback* y el *awareness*. Lo anterior puede ser debido a que HCI se ha enfocado en proveer mecanismos, técnicas, herramientas en el nivel de la interacción con el sistema, pero no tanto en el nivel independiente de la tecnología.

Esto, a pesar de que en la imagen representativa de los elementos que involucra el HCI, aparece el "Uso y Contexto" y el "Trabajo y Organización Social (U1)" (Figura 5.13), lo que involucra la interacción entre las personas y la organización.

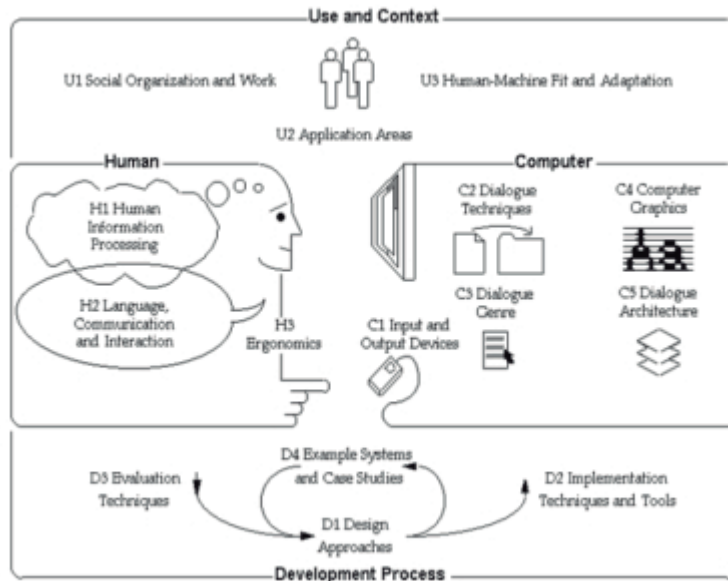


Figura 5.13. Imagen que resume los elementos relacionados a HCI (Granollers, 2004).

Es posible concluir también que la evaluación debería haberse diseñado no solamente desde el punto de vista de los elementos notacionales, sino también desde el punto de vista de los diagramas; esto debido a que desde la perspectiva del diseño, la información es capturada a través de diagramas. Adicionalmente, en el contexto faltaría indicar si los elementos notacionales a evaluar tienen o no, relaciones o conexiones, lo que hace parte también de los diagramas.

El hecho de que se seleccionen los mismos elementos de modelado para representar los clasificadores <Tarea que hace el Worker> y <Tarea que hace el Usuario>, permite entender que la captura de información a nivel de la organización, de la interacción entre las personas, está co-relacionado con la captura de información a nivel del sistema informático. Es decir, existe una trazabilidad y a la vez una separación entre estos dos niveles de abstracción. De hecho, por esta razón se definen también en la TxA tareas de comunicación (*Inter-Acción*) en el nivel de Negocio.

Tanto en el nivel de negocio como en el nivel de sistema, se observa que los aspectos relacionados con la Labor se capturan a través de la técnica de Casos de Uso. Como se ha visto en la sección 5.1, es necesario incorporar aspectos de HCI a este tipo de artefactos.

5.3 DEFINICIÓN DE MARCOS DE DESARROLLO

En esta sección se explica cómo a partir del conjunto de clasificadores de la TxA, se definen un conjunto de elementos que permiten capturar la información a utilizar en la especificación y modelado de sistemas interactivos. Dicho conjunto de elementos conforman el Roadmap propuesto a partir de la infraestructura de CIAF: Método, Notación y Herramienta.

5.3.1 Adaptación del Método de Desarrollo de la Interfaz de Usuario

La propuesta de desarrollo de la interfaz de usuario en la que se basa este Marco de Desarrollo, se centra en combinar el diseño de las interfaces basadas en modelos de datos y de tareas, de ahí su nombre: TD-MBUID (*Task & Data – Model Based User Interface Development*) (Giraldo, 2010). En la Figura 5.14 se presenta el flujo de desarrollo de la interfaz de usuario en TD-MBUID; el cual se lleva a cabo en dos niveles de modelado. Inicialmente, se analizan las tareas de alto nivel del negocio y los objetos del dominio que las soportan. Estas actividades y objetos, aunque son independientes de la tecnología, permiten entender los modelos mentales de los usuarios para desarrollar la interfaz de usuario de negocio a partir de los datos que están siendo utilizados dentro del contexto de cada proceso. El objetivo del modelado de la interfaz de usuario de negocio es identificar los formularios, generalmente en papel, que soportan la entrada de información a los procesos de negocio de una manera realista e independiente de la tecnología; adicionalmente, se identifica la navegación que ocurre entre dichos formularios. Posteriormente, se analizan las tareas interactivas y los objetos del sistema que las soportan, para identificar el diálogo entre el usuario y el computador. Este diálogo se lleva a cabo mediante un conjunto de interfaces que son generadas a partir de una especificación abstracta, hasta obtener una especificación final ejecutable.

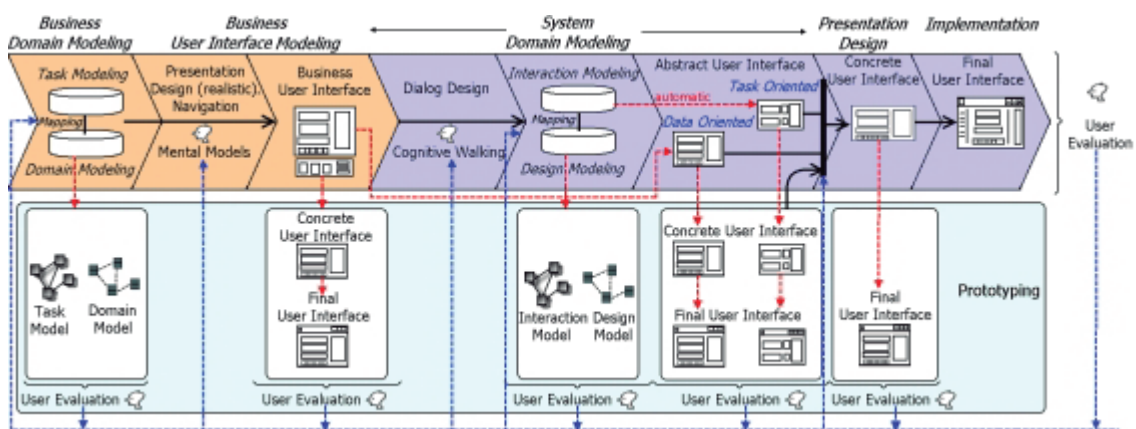


Figura 5.14. Flujo de desarrollo de la interfaz de usuario en TD-MBUID (Giraldo, 2010).

El flujo de desarrollo propuesto se muestra en la Figura 5.15. Esta presentación obedece a la especificación metodológica utilizando el lenguaje para especificación de procesos SPEM (OMG, 2008a).

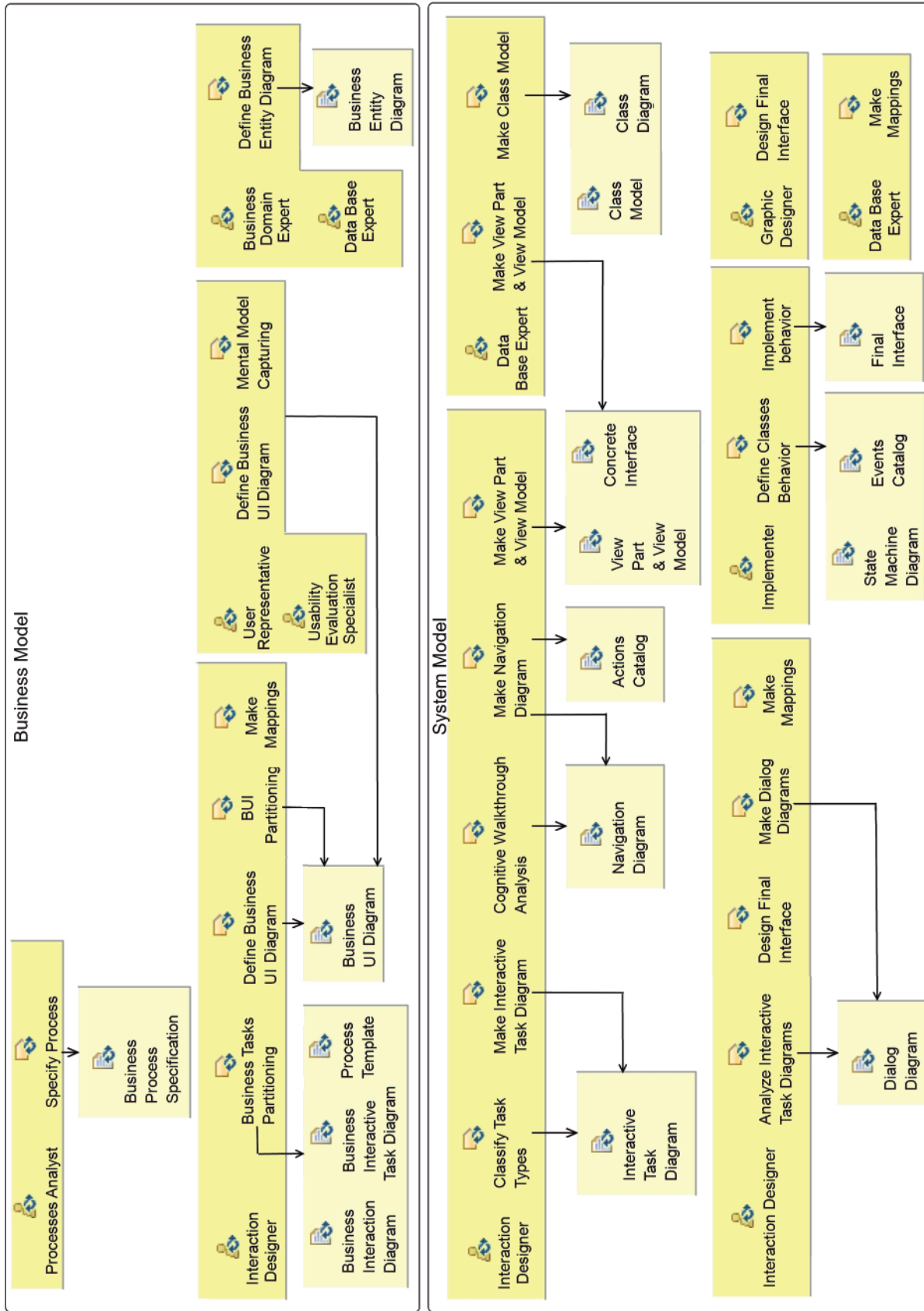
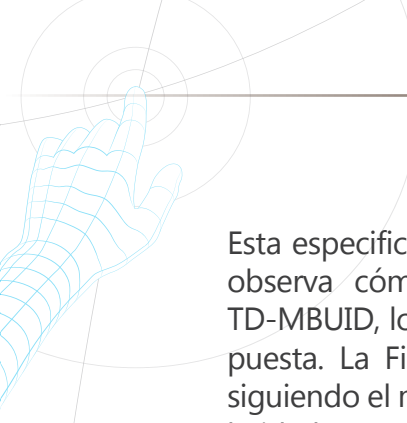


Figura 5.15. Mapa de ruta propuesto, especificación en SPEM.



Esta especificación ofrece una vista de actividades, roles y artefactos y sus relaciones. Se observa cómo se conserva la separación en niveles de abstracción plasmada en TD-MBUID, lo cual también está acorde con la estructura taxonómica de la actividad propuesta. La Figura 5.16 muestra una vista detallada de los artefactos que se generan siguiendo el mapa de ruta propuesto. Se observa cómo se conservan los niveles de granularidad para cada nivel de abstracción, Negocio y Sistema. En este flujo de desarrollo se agregan algunos artefactos, se cambia a una vista vertical del mapa de ruta y algo para resaltar es que se enfoca en trabajar el particionamiento tanto de los modelos de tareas, como de las interfaces en todos los niveles. La navegación por el roadmap inicia con la definición de un proceso de negocio que se desee implementar.

A continuación se describen las actividades que se realizan a medida que se sigue el mapa de ruta:

5.3.1.1 Modelo de análisis de negocio (Business analysis model)

En este paso se analiza el dominio del proceso para identificar al cliente, al worker, las tareas de interacción entre personas y los datos que se manipulan. El resultado de este paso es una descripción detallada del proceso que se desea desarrollar con la metodología propuesta.

En este nivel, se captura información del proceso, de manera generalizada e independiente de la tecnología. Se trata de entender el contexto del proceso y los datos que están asociados a ese contexto. Se utilizan técnicas como entrevistas, observación de los usuarios, para capturar su modelo mental sobre cómo comprenden el proceso y qué datos deberían mostrarse en una interfaz de usuario de negocio, en papel y sin botones o widgets concretos, (*Business User Interface, BUI*). Se utilizan también los conocimientos de las leyes de Gestalt y las diferentes formas existentes de presentar los datos. En este nivel, también podrá llevarse a cabo pruebas de usabilidad simplificadas y verificación de accesos a elementos de la BUI. Esto puede dar lugar a cambios en la descripción del proceso de negocio.

5.3.1.2 Modelo de Diseño de Negocio (Business Design Model)

Este paso tiene como resultado la realización del proceso (*Process Realization*). Se realiza el diseño de las interacciones entre las personas que intervienen en el proceso, cliente y worker. También de las relaciones entre dichas interacciones y entre los datos que son manipulados. Este diseño tiene dos niveles de granularidad, alto y bajo. El nivel bajo contiene 3 artefactos de diseño: **1. Plantilla de Proceso** (*Process Template*), que a su vez contiene dos diagramas: Diagrama de Interacción de negocio (*Business Interaction Diagram*) y Diagrama de Tareas Interactivas de Negocio (*Business Interactive Task Diagram*), **2. Diagrama de Interfaz de Usuario de Negocio** (*Business User Interface Diagram*) y **3. Diagrama de Entidades de Negocio** (*Business Entity Diagram*). El nivel alto se encuentra contenido en el diagrama de tareas interactivas de negocio, donde cada tarea se encuentra desglosada de acuerdo a la especificación del proceso.

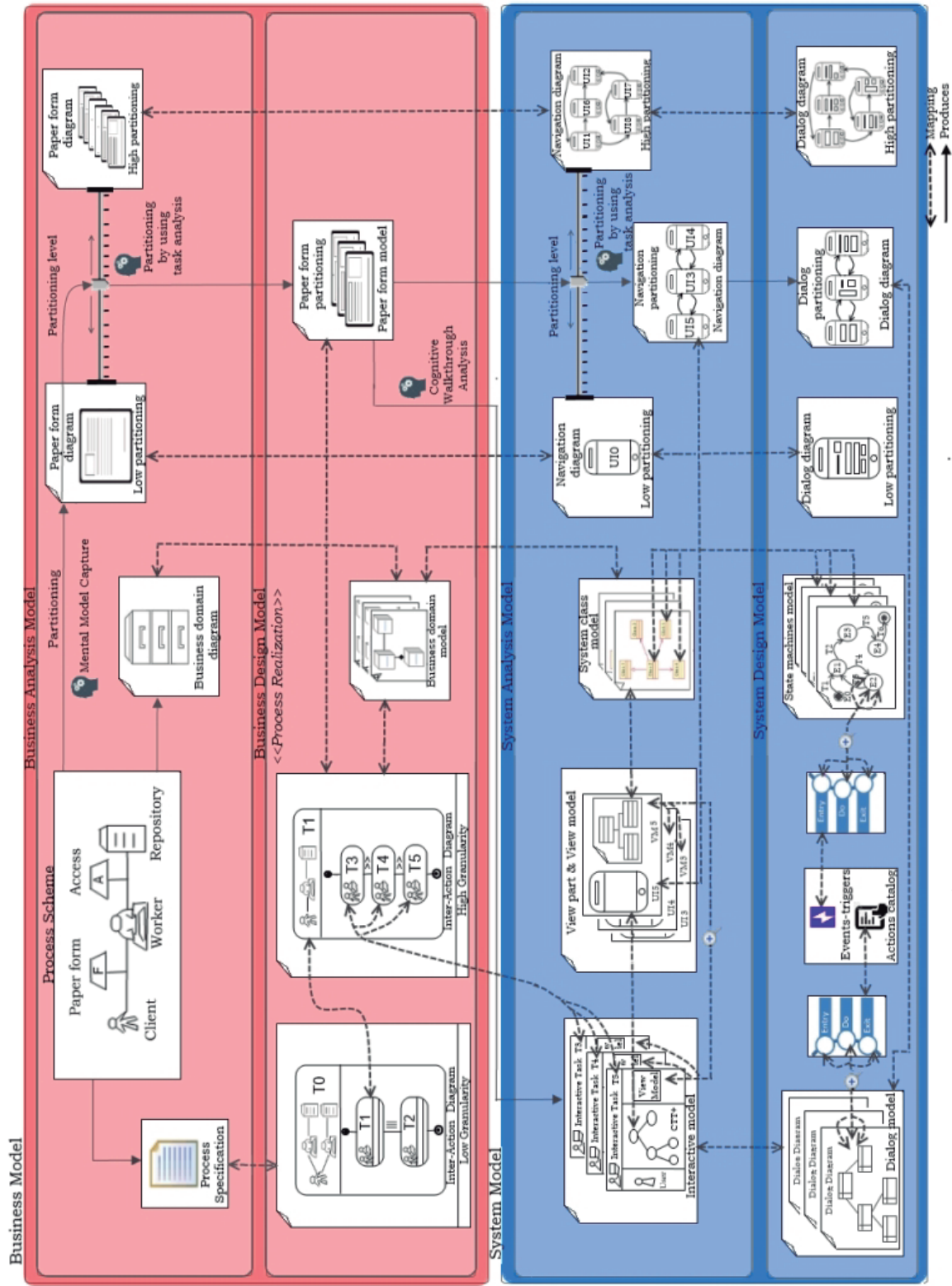
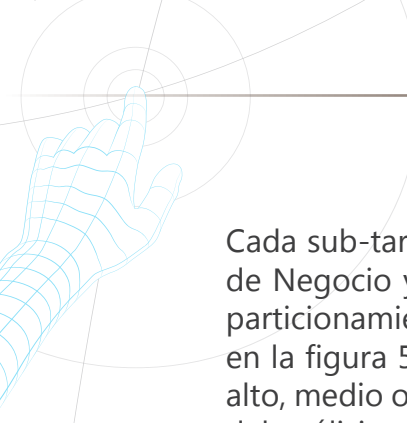


Figura 5.16. Mapa de ruta propuesto, especificación detallada de artefactos.



Cada sub-tarea en este nivel tiene un mapping a un Diagrama de Interfaz de Usuario de Negocio y a un Diagrama de Entidades de Negocio y de esta forma se plasma el particionamiento realizado a la BUI a partir del análisis de las tareas. Como se muestra en la figura 5.16, parte superior derecha, puede resultar un nivel de particionamiento alto, medio o bajo. El interés es obtener un equilibrio en dicho particionamiento a partir del análisis y pruebas realizadas en el nivel del Modelo de Análisis del Negocio.

5.3.1.3 Modelo de Análisis de Sistema (System Analysis Model)

En esta fase se realiza el diseño interactivo, que muestra el modo en el que el usuario puede interactuar con el sistema. A partir de los diagramas de Interfaces de Usuario de Negocio, se utiliza la técnica de recorrido cognitivo para cada BUI. Se identifican y clasifican entonces las tareas que son propias de los usuarios, las tareas interactivas y las tareas que serán llevadas a cabo por el computador. El resultado es un conjunto de diagramas de tareas interactivas a nivel de sistema, donde cada diagrama de tareas corresponde con cada tarea interactiva definida en el nivel alto de granularidad a nivel de negocio.

La técnica de recorrido cognitivo ayuda a definir el modelo de navegación para el proceso analizado. Como se observa en la figura 5.16, mitad derecha, el diagrama de navegación resultante, corresponde con el nivel de particionamiento realizado para la interfaces de usuario de negocio. Igualmente, los diagramas de clases de sistema tendrán un mapping con los diagramas de entidades de negocio (Figura 5.16 mitad). A partir de los diagramas de tareas interactivas y de los diagramas de clases, a nivel de sistema, se obtiene la vista de partes de interfaz y partes del modelo de datos (*View Part* y *View Model*). Cada uno de estos diagramas muestra una pareja entre un trozo de la interfaz y la vista de datos que corresponde con dicha interfaz.

5.3.1.4 Modelo de Diseño de Sistema (System Design Model)

En este paso se crean los Diagramas de Diálogo, a partir de los diagramas de tareas interactivas a nivel de sistema. Se obtienen también los diseños detallados de las interfaces de usuario. Para ello, se tienen en cuenta reglas de diseño y ergonomía y requerimientos de usabilidad. En los Diagramas de Diálogo, se definen los controles que necesitará el usuario para llevar a cabo sus tareas. También se decide el modo de llevar a cabo las acciones sobre la interfaz identificadas en el paso anterior, por ejemplo, mediante botones, menús, arrastrar y soltar, etc.

En este nivel se diseñan los Diagramas de Estados que están asociados a las clases de los Diagramas de Clases de Sistema. Estos diagramas de estados definen los ciclos de vida de dichas clases. El modelo de navegación y los diagramas de estados están relacionados a través de un catálogo de acciones y un catálogo de eventos, respectivamente. Estos catálogos definen las funciones necesarias de navegación necesarias de navegación, el menú de navegación, la información y la funcionalidad que ejecuta el proceso modelado.



5.3.1.5 Prototipado y evaluación (Prototype & Evaluation)

Al igual que en TD-MBUID, el prototipado y la evaluación es una etapa que se ejecuta de manera transversal. Se evalúa la usabilidad de las soluciones diseñadas y se comparan con las metas de usabilidad. Después de aplicar las pruebas de usabilidad a los prototipos, se evalúan los problemas y se corrigen en la medida de lo posible. Se puede llevar a cabo una evaluación heurística para apuntar a los posibles problemas.

5.3.2 Contenido Metodológico detallado de la propuesta

El contenido detallado de la propuesta se centra en describir los artefactos presentados en el mapa de ruta (Figura 5.16) por niveles de abstracción. Se describen también los roles que intervienen en el desarrollo de la interfaz de usuario.

5.3.2.1 Modelo de Negocio (Business Model)

En este nivel se generan los siguientes artefactos:

Diagrama de Plantilla de Proceso (*Process Template Diagram*): Contiene la especificación de la Interacción entre los actores del negocio (*Business Interaction*) y la especificación de las tareas interactivas de negocio (*Business Interactive Task*). El primero, describe la interacción entre el Cliente y el (los) Worker (s) a través de formularios en papel (*interaction*) y también la interacción con las entidades del negocio (*access*). El segundo, describe la secuencia de tareas interactivas de negocio. Se soportan dos niveles de granularidad para dichas tareas interactivas, es decir, es posible especificar Tareas que se componen de Tareas Individuales, Tareas Cooperativas y/o Tareas de Comunicación.

Un modelo de tareas describe un ordenamiento lógico y temporal de las tareas que son realizadas por los usuarios en la interacción con un sistema. Un modelo de tareas, a nivel de negocio, describe tanto las tareas actuales de los usuarios como la forma en que se pueden realizar en el futuro. Los elementos individuales en un modelo de tareas, representan las acciones específicas que el usuario puede llevar a cabo (Giraldo, 2010).

Diagrama de Interfaz de Usuario de Negocio (*Business User Interface Diagram*): Contiene la especificación de una interfaz de papel o formulario que soporta el proceso en cuestión. El número de diagramas de este tipo se encuentra determinado por número de tareas simples especificadas en la Plantilla de Proceso. Es decir, cada uno de estos diagramas representa una partición de la interfaz que soporta el proceso de acuerdo a la especificación de las tareas de negocio. Existe una relación de *mapping* entre la Plantilla de Proceso y cada Diagrama de Interfaz de Usuario de Negocio.

La Interfaz de usuario de negocio (*Business User Interface*) corresponde a una parte del modelo de la interfaz de usuario; es la información que el usuario debe imaginar que está detrás de la pantalla física - en la parte posterior del sistema (Lauesen, 2005). Normalmente se dibuja esta interfaz, en términos de elementos de interfaz de usuario concretos (pantallas en papel) debido a que se asemejan a las pantallas finales en el computador, con detalles gráficos y con contenidos realistas de los datos. La

organización de la interfaz de usuario de negocio se realiza preservando las diferentes leyes de Gestalt: la ley de proximidad, ley de cierre, la ley de la buena continuación y la ley de la similitud.

Diagrama de Entidades de Negocio (*Business Entity Diagram*): Contiene la especificación del conjunto de entidades necesarias para la realización del proceso en cuestión. El número de diagramas de este tipo se encuentra determinado por número de tareas simples especificadas en la Plantilla de Proceso. Es decir, cada uno de estos diagramas representa una partición del conjunto de entidades que requiere el proceso de acuerdo a la especificación de las tareas de negocio. Existe una relación de *mapping* entre la Plantilla de Proceso y cada Diagrama de Entidades de Negocio.

El modelo de dominio captura la semántica del contexto del sistema y define los requisitos de información para el desarrollo de la interfaz de usuario. Un modelo de dominio describe los objetos de un dominio de aplicación. Especifica los datos que las personas utilizan, relacionados con las entidades del mundo real y las interacciones tal como son entendidas por los usuarios en relación con las acciones que son posibles de realizar sobre estas entidades. El análisis del dominio es parte de la mayoría de los enfoques de desarrollo, y no es algo específico a considerar en el diseño de la interfaz de usuario. Los objetos del dominio son considerados como instancias de clases que representan los conceptos que son manipulados y que son totalmente independientes de la forma en que se muestran en la pantalla y cómo se almacenan en el computador.

5.3.2.2 Modelo de Sistema (*System Model*)
























En este nivel se generan los siguientes artefactos:

Patrón de Presentación (*Presentation Pattern*): Contiene el conjunto de diagramas necesarios para especificar, a nivel del sistema interactivo, cada una de las tareas simples especificadas en la Plantilla de Proceso, a nivel de negocio. Cada una de estas tareas simples tiene una trazabilidad hacia los siguientes diagramas: **1. Diagrama de Tareas Interactivas** (*Interactive Task Diagram*), **2. Diagrama de Diálogo** (*Dialog Diagram*), **3. Vista de la Interfaz Concreta** (*View Part Diagram*), **4. Vista del Modelo de Datos** (*View Model Diagram*), **5. Modelo de Dominio** (*Domain Model Diagram*), **6. Diagramas de Estados** (*StateChart Diagram*). Para cada una de las tareas simples especificadas en la Plantilla de Proceso, es preciso crear un Patrón de Presentación.

En el **Diagrama de Tareas Interactivas** se conserva la estructura y relaciones temporales de la notación CTT. Se agregan atributos a las tareas interactivas para denotar si expresan un cierre de tarea, una interacción con feedback o una interacción con awareness. Además este diagrama indica explícitamente el usuario que realiza las tareas.

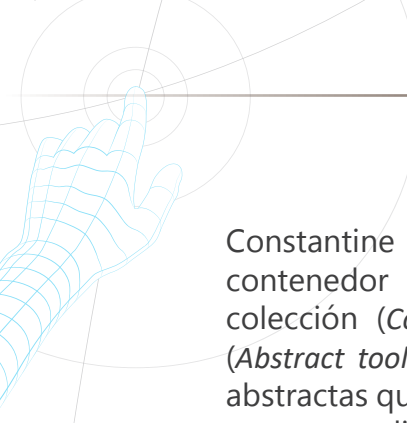
El **Diagrama de Diálogo** conserva la estructura del Diagrama de Tareas, especificando además para cada tarea, el tipo de Componentes de Interacción Abstractos (AIC) asociados a la tarea. Estos componentes de interacción representan una expresión canónica de la renderización y la manipulación de los conceptos del dominio y las

funciones, de una manera que sea lo más independiente posible de la modalidad (auditiva, visual, táctil, etc) y de la plataforma informática (Limbourg, 2004). La Figura 5.17 presenta los Componentes de Interacción Abstractos presentados por Contantine (Constantine et al, 2003).

			TOOLS
SYMBOL	INTERACTIVE FUNCTION	EXAMPLES	
	action/operation*	Print symbol table, Color selected shape	
	start/go/to	Begin consistency check, Confirm purchase	
	stop/end/complete	Finish inspection session, Interrupt test	
	select	Group member picker, Object selector	
	create	New customer, Blank slide	
	delete, erase	Break connection line, Clear form	
	modify	Change shipping address, Edit client details	
	move	Put into address list, Move up/down	
	duplicate	Copy address, Duplicate slide	
	perform (& return)	Object formatting, Set print layout	
	toggle	Bold on/off, Encrypted mode	
	view	Show file details, Switch to summary	
			MATERIALS
SYMBOL	INTERACTIVE FUNCTION	EXAMPLES	
	container*	Configuration holder, Employee history	
	element	Customer ID, Product thumbnail image	
	collection	Personal addresses, Electrical Components	
	notification	Email delivery failure, Controller status	
			ACTIVE MATERIALS
SYMBOL	INTERACTIVE FUNCTION	EXAMPLES	
	active material*	Expandable thumbnail, Resizable chart	
	input/accepter	Accept search terms, User name entry	
	editable element	Patient name, Next appointment date	
	editable collection	Patient details, Text object properties	
	selectable collection	Performance choices, Font selection	
	selectable action set	Go to page, Zoom scale selection	
	selectable view set	Choose patient document, Set display mode	

* Generic component, can substitute for any specialization.

Figura 5.17. Componentes abstractos canónicos presentados por Constantine (Constantine et al, 2003)



Constantine propone 4 tipos de Materiales Abstractos (*Abstract Materials*): un contenedor genérico (*Generic container*), un solo elemento (*single Element*), una colección (*Collection*), y una notificación (*Notification*). Las herramientas abstractas (*Abstract tools*) incluyen operaciones y acciones. Las operaciones son herramientas abstractas que operan sobre materiales y las acciones son herramientas abstractas que causan o disparan alguna acción. Los Materiales Activos son los componentes abstractos que exhiben características de ambos, herramientas y materiales.

Para la **Vista de Interfaz**, la **Vista del Modelo de Datos** y el **Modelo de Dominio**, se utiliza el concepto del patrón MVVM (*Model-View-View Model Pattern*) (Microsoft, 2017). En este caso, la Vista representa la estructura, despliegue y la apariencia de la Interfaz de Usuario Concreta con sus componentes. El Modelo es una representación de los datos que soportan la especificación de la Interfaz Concreta. Tiene trazabilidad con el Diagrama de Entidades de Negocio. Por último, la Vista del Modelo representan los datos que tienen un mapeo directo hacia la Interfaz Concreta o Vista.

Básicamente, la **Vista de Interfaz** es un modelo de la interfaz de usuario que permite especificar el aspecto y el comportamiento de la interfaz de usuario por medio de elementos que pueden ser percibidos por los usuarios (Limbourg, 2004). Se trata de una especificación de la interfaz de usuario en términos de objetos de interacción concretos (CUI). Un objeto de interacción concreto es un componente de la interfaz que puede ser manipulado o percibido por el usuario (por ejemplo, una ventana, un botón, una casilla de verificación, etc). Aunque los objetos de interfaz concreta están más relacionados con los *widgets*, siguen siendo independientes de la representación real o *rendering*.

Los **Diagramas de Estados** representan el comportamiento de las Clases que conforman el Modelo del Sistema. Cada Clase puede tener asociado uno o varios diagramas de estados.

Diagrama de Navegación (*Navigation Diagram*): El diagrama de Navegación presenta la interacción existente entre los patrones de presentación, mediante las acciones que permiten conmutar el sistema interactivo desde una tarea hacia las demás tareas interactivas. Sirve como un elemento de orquestación entre la especificación de los modelos mencionados.

En el mapa de ruta no se encuentra especificada la *Interfaz de usuario final* (*Final User Interface*) porque es la interfaz de usuario en funcionamiento, es decir, cualquier interfaz de usuario que se ejecuta en una plataforma informática, ya sea por interpretación (por ejemplo, a través de un navegador Web) o por ejecución (por ejemplo, por la compilación de su código en un entorno de desarrollo interactivo). La interfaz de usuario final tiene dos posibles representaciones: el propio código que la define y la *renderización* o visualización de la misma (Limbourg, 2004).

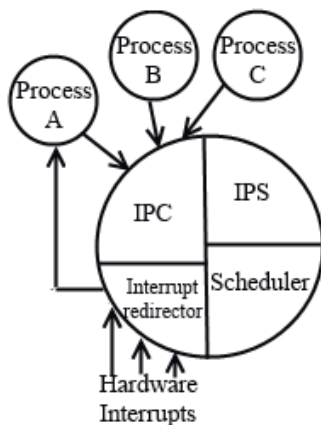
5.3.2.3 Modelo de Ejecución (*Execution Model*)

El modelo de ejecución para esta propuesta está soportado básicamente por dos

componentes. Un ejecutivo multitarea en tiempo real (EMTR) (Giraldo, 2000), (Giraldo et al, 2004), (Giraldo et al, 2001) y un componente para la ejecución de máquinas de estado de UML (Villegas et al, 2009).

Ejecutivo Multitarea en Tiempo Real (EMTR)

El EMTR provee los servicios necesarios para planificación, comunicación y sincronización entre procesos, administración de interrupciones y excepciones (Figura 5.18).



En toda aplicación de tiempo real, el control del tiempo es fundamental. El EMTR soporta sólo un tipo de primitivas para manejo de tiempo. Delay() es una de las primitivas más usadas en todo sistemas operativo.

● **Figura 5.18.** Estructura del EMTR (Giraldo, 2000)

El módulo IPS es el encargado de proporcionar las primitivas básicas de sincronización de todo ejecutivo multitarea en tiempo real. Permite la protección de recursos (exclusión mutua) y la sincronización de dos o más procesos. El módulo IPC es el encargado de proporcionar las primitivas básicas de comunicación, aunque cuenta básicamente con dos directivas de comunicación, éstas permiten una variedad de implementaciones de tal manera que reemplacen algunas existentes en otros kernel tales como Rendez-vous.

El planificador (scheduler) de EMTR es considerado como un proceso más del sistema, es inicializado por medio de la primitiva `Initialise()` con una prioridad cero. Aunque es considerado un proceso más, el planificador nunca es planificado, y su código se ejecuta en cada Tick del reloj. El manejador de procesos es parte del conjunto del ejecutivo, es responsable de crear nuevos procesos en el sistema y administrar los recursos más fundamentales asociados con un proceso. Los manejadores de interrupción soportan el sistema de interrupciones hardware del computador. Ellos reaccionan a las interrupciones hardware y manejan las transferencias de bajo nivel de datos entre el computador y los dispositivos externos.

EMTR desarrolla algunos chequeos de validación durante las invocaciones de las primitivas del programa ejecutivo. Cuando una operación anormal se detecta, la ejecución del código de la tarea es redirigida al manejador de excepciones.

Ejecución de máquinas de estado

El componente UML Semantics contiene la implementación del metamodelo de UML v.1.1 (Rational, 1997), específicamente la parte de los elementos del comportamiento de los objetos que hace referencia a los paquetes *Foundation* y *BehavioralElements*. El propósito de este componente en forma de librería es básicamente brindar las funciones de la semántica bien definida del comportamiento de las aplicaciones modeladas en lenguaje UML, para la implementación de aplicaciones sobre la plataforma del Ejecutivo Multitarea en Tiempo Real (EMTR) partiendo del diseño. En el **Anexo F** se presenta la descripción detallada del modelo de diseño para este componente. La especificación del modelo de ejecución, permite cumplir con la **Estrategia 14**, planteada en la Sección 1.2.

La Figura 5.19 muestra la relación entre los componentes EMTR y UML Semantics, exponiendo las interfaces más relevantes que brinda el ejecutivo.

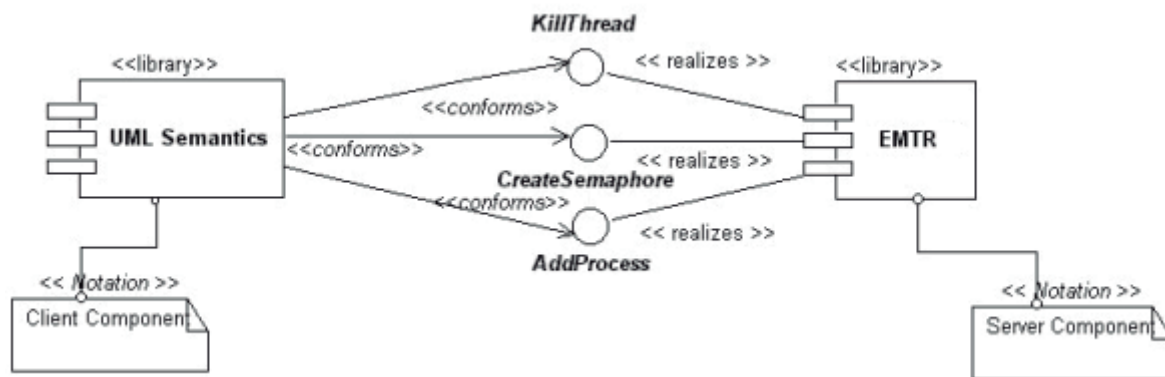


Figura 5.19. Relación entre los componentes EMTR y UML Semantics (Villegas et al, 2009).

A nivel de una aplicación construida por el usuario, se agregaría al diagrama anterior los componentes "archivo" de la aplicación, que también hacen uso de varias interfaces brindadas por el kernel.

Adaptación del modelo de ejecución

Específicamente, para este trabajo, ha sido necesaria la realización de una adaptación de los componentes mencionados. Primero, porque estos componentes estaban implementados en lenguaje C y C++, sobre el sistema operativo D.O.S, y segundo, porque es necesario utilizar la lógica de las máquinas de estado para ejecutar el modelo de navegación diseñado si se sigue el roadmap propuesto.

El proceso de adaptación abarca la traducción de todo el código fuente de los componentes, de lenguaje C y C++ a lenguaje Java. Este proceso de traducción es un poco tedioso cuando se encuentra dentro del código fuente la implementación con punteros, elementos que no están disponibles en el lenguaje Java. El proceso de

adaptación incluye también la integración de los componentes ya traducidos en lenguaje Java, a la herramienta de software CIAT, enriquecida con los editores que permiten la diagramación y especificación de los modelos que se describieron en el roadmap propuesto, (Figuras 5.15 y 5.16).

El componente para la ejecución de máquinas de estado se encarga de atender los eventos y acciones generados a partir de la interacción de los usuarios con las interfaces, invocando los servicios del EMTR. Se ejecutan también las acciones de entrada, acciones de salida y las que se deben ejecutar dentro de cada estado (*entry*, *exit*, *do*), ya sea cuando se activa una máquina de estado, asociada a una clase de sistema, o cuando se activa una ventana de diálogo, asociada al modelo de navegación. Esta secuencia de ejecución se mostrará en detalle con el caso de estudio, Sección 5.3.5. La adaptación del modelo de ejecución, permite cumplir con las **Estrategias 15 y 16**, planteadas en la Sección 1.2.

5.3.2.4 Roles

Los roles que intervienen en el desarrollo de la interfaz de usuario son (Göransson et al, 2003):

- *Diseñador de usabilidad (Usability Designer)*: Es el jefe o líder de campo de la usabilidad. La persona que actúe en este rol debe ser un experto en HCI con antecedentes en ciencias de la computación y experiencia en el proceso de diseño centrado en el usuario. Se encarga de la planificación y la gestión de las actividades relacionadas con la usabilidad, así como la realización de algunas de ellas.
- *Especialista en estudios de campo (Field Study Specialist)*: Este rol debe tener experiencia en el uso de diferentes métodos y técnicas de usabilidad para la clasificación de grupos de usuarios y la comprensión de las necesidades de los usuarios. Normalmente esta persona debería ser capaz de planificar, ejecutar y analizar estudios de usuarios. Es preferible que posea un conocimiento general en HCI.
- *Diseñador de interacción (Interaction Designer)*: Se encarga de los aspectos conceptuales, la interacción y el diseño detallado. Esto incluye: la creación del esquema general de la interacción y la estructura; la definición de la dinámica de la interacción del usuario, las rutas de navegación y la estructura de los datos. Se necesita experiencia y una combinación de conocimientos en HCI y diseño.
- *Diseñador gráfico (Graphic Designer)*: Este rol es responsable de la configuración visual de la interfaz de usuario y trabaja en estrecha colaboración con el diseñador de interacción. Requiere experiencia en diseño creativo y de medios interactivos.

- *Especialista en la evaluación de usabilidad (Usability Evaluation Specialist)*: Este rol se encarga de la planificación, preparación, ejecución y presentación de informes de las evaluaciones de usabilidad. Debe ser un experto en la realización de evaluaciones en laboratorios de usabilidad, así como en el campo. Los antecedentes de este rol se ubican en el área de HCI con un enfoque en ciencias del comportamiento humano.
- *Experto en el Dominio del Negocio (Business Domain Expert)*: Este rol es responsable de contextualizar a los demás involucrados en el desarrollo de la interfaz, acerca del proceso de negocio. Conoce su contexto, los conceptos de dominio de negocio relevantes en el proceso que se pretende desarrollar y cómo se relacionan dichos conceptos.
- *Experto en Bases de Datos (Data Base Expert)*: Este rol se encarga de definir las entidades de negocio, sus relaciones y demás características necesarias para la construcción de los modelos de entidades de negocio y modelos de clases de sistema necesarios para el desarrollo del proceso. Es preferible que posea un conocimiento general en HCI.

5.3.3 Modificaciones a la notación usada en TD-MBUID y definición de algunos elementos de modelado

El objetivo de esta sección es presentar algunas mejoras llevadas a cabo en la notación utilizada en la propuesta TD-MBUID para el desarrollo de la interfaz de usuario y presentar la propuesta de notación para algunos elementos de modelado que se utilizan al seguir el roadmap según la Figura 5.16.

Las mejoras en la notación se desarrollan pensando en los siguientes objetivos:

1. Adaptar la notación para que sea más consecuente con la definición de la Taxonomía de la Actividad, presentada en el capítulo 3. De esta forma, se facilita la conexión con el desarrollo de la interfaz de usuario.
2. Enriquecer la información que se captura en los elementos de modelado para complementar los modelos adaptados desde TD-MBUID (sección 5.3.2)
3. Adaptar el metamodelo de CIAF de tal forma que pueda implementarse una herramienta de modelado que facilite la edición y la generación de código de la aplicación a partir de los modelos presentados en la sección 5.3.2.

La notación o sintaxis concreta de un lenguaje de modelado, es la representación textual o gráfica del lenguaje que se diseña y es la que permite implementar modelos en dicho lenguaje. De esta forma, la sintaxis concreta facilita o dificulta el entendimiento de los diagramas y la organización de los modelos que describen un sistema (Giraldo, 2010). Las actualizaciones que se llevan a cabo en la notación usada en TD-MBUID están orientadas a mejorar la expresividad y la estructuración de los

modelos, de cara a facilitar el modelado de la actividad en sistemas interactivos para el desarrollo de la interfaz de usuario.

En primer lugar, se mantiene el concepto de diagrama en el marco conceptual (sintaxis abstracta) de CIAF, con lo cual se identifica la asignación de elementos de modelado a dichos diagramas. Esta organización se establece conforme a los distintos niveles de abstracción del marco de desarrollo. Se consideran entonces elementos de modelado a nivel de negocio y a nivel de sistema.

La modificación más significativa que se realiza a la notación utilizada en TD-MBUID, es en las tareas de inter-acción, como se ilustra en la Figura 5.20. Los demás cambios sólo se hacen a nivel de iconografía y se destaca que se adicionan algunos diagramas que no los tiene definidos TD-MBUID.

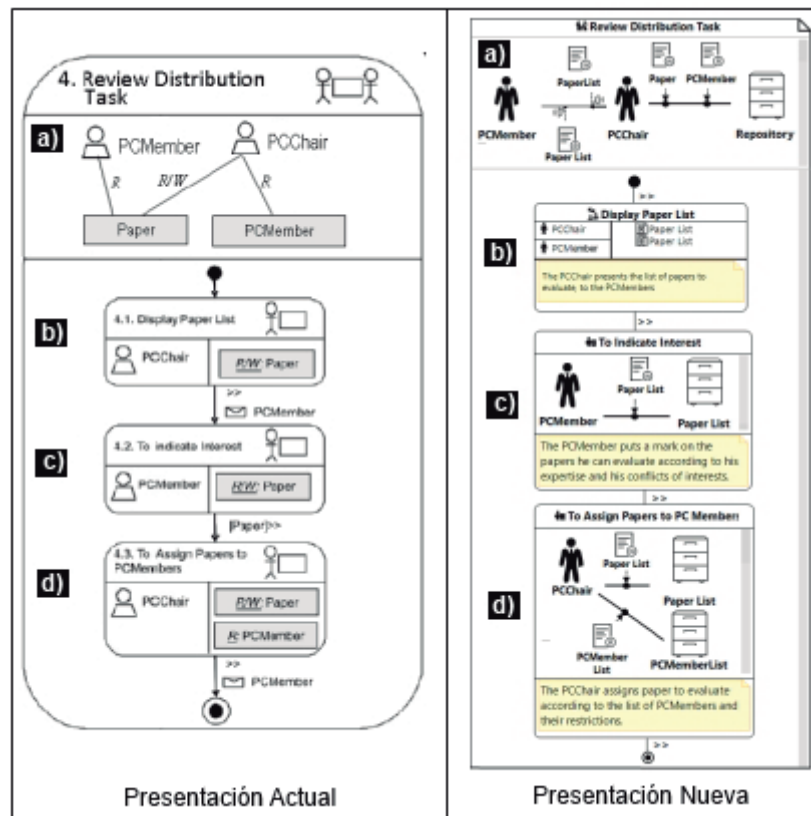


Figura 5.20. Aspecto de las tareas de Inter-Acción de TD-MBUID. Propuesta de modificación, Tarea interactiva de negocio.

En la Figuras 5.21 a 5.24 es posible apreciar que se ha logrado enriquecer la especificación de la interacción, entre los roles y también entre los roles y las entidades de negocio. El modelado de esta interacción provee la siguiente información:

1. Se especifica claramente cuál es el medio a través del cual se establece la comunicación entre los roles. Esto es útil en el diseño de las interfaces de usuario de negocio (Business User Interfaces).

- Se especifica claramente cuál es el modificador de acceso entre cada rol y cada entidad. Esto es útil en el diseño de la interfaz para configurar los accesos que se pueden llevar a cabo mediante los componentes de la misma.

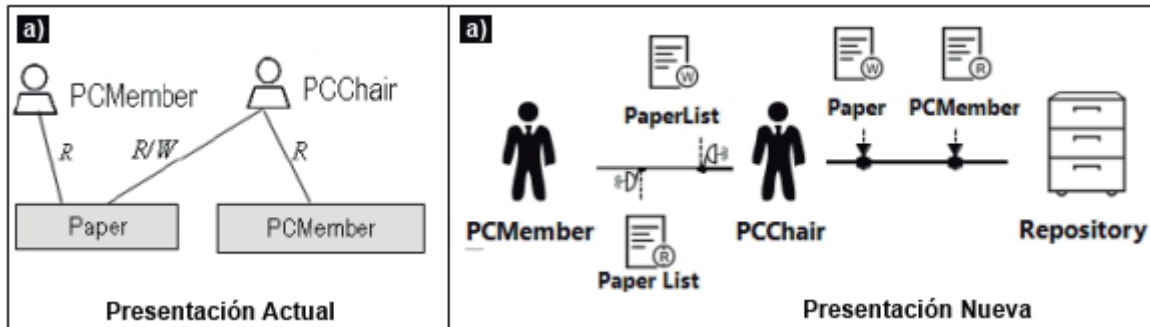


Figura 5.21. Detalle de presentación actual y presentación nueva, Inter-acción de negocio

- Al igual que en TD-MBUID, se describe de forma visual con qué entidades interactúa cada rol mediante las relaciones y modificadores de acceso. Se puede documentar cada relación (mediante un campo de descripción) para especificar adicional información relativa a las necesidades del usuario en la interacción con ese objeto de datos. Dicha información puede hacer referencia al contexto específico de uso del sistema.

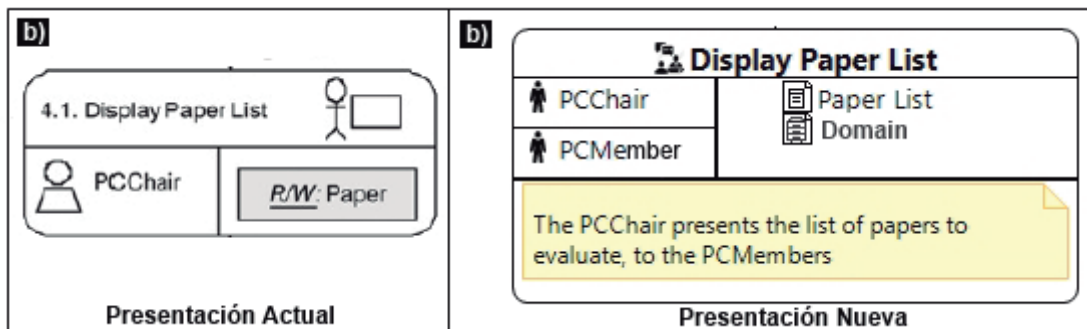


Figura 5.22. Detalle de presentación actual y presentación nueva para tarea de comunicación

- Se muestra gráficamente cuál es la relación entre los objetos de datos, mediante los Diagramas de Entidades de Negocio. Entender la estructura de la información relacionada con una tarea de alto nivel de granularidad, como las de Inter-acción, facilita la especificación de la estructura de la interfaz de usuario. Se pueden identificar, dentro de las relaciones de objetos, patrones conocidos que pueden conducir a patrones concretos de diseño de la estructura de la interfaz de usuario.
- En el Caso de Estudio, se mostrará gráficamente cuál es la distribución de los datos requeridos para llevar a cabo cada tarea, mediante los Diagramas de Interfaces de Usuario de Negocio. Al igual que con los Diagramas de Entidades

de Negocio, es posible identificar, dentro de las interfaces de negocio, patrones conocidos que pueden conducir a patrones concretos de diseño de la estructura de la interfaz de usuario concreta.

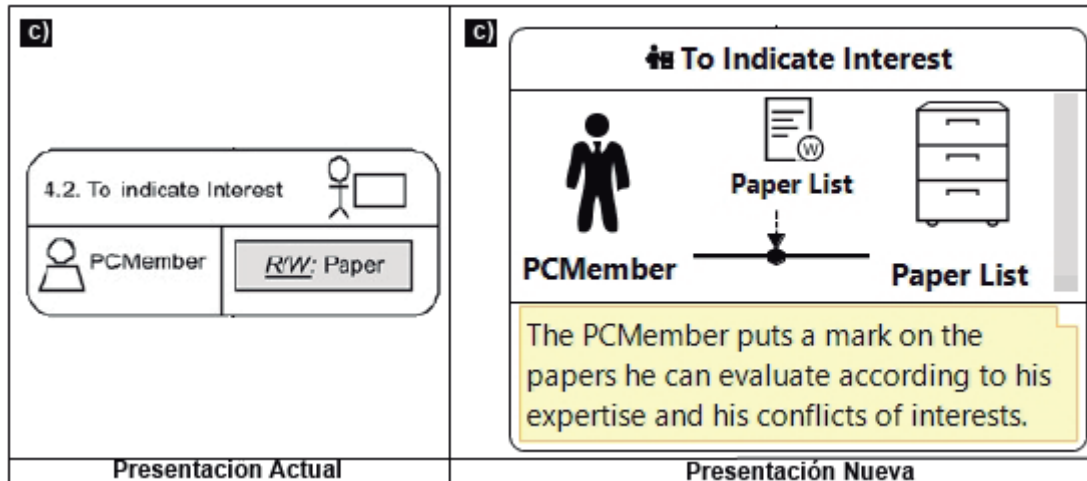


Figura 5.23. Detalle de presentación actual y presentación nueva, tarea individual simple c)

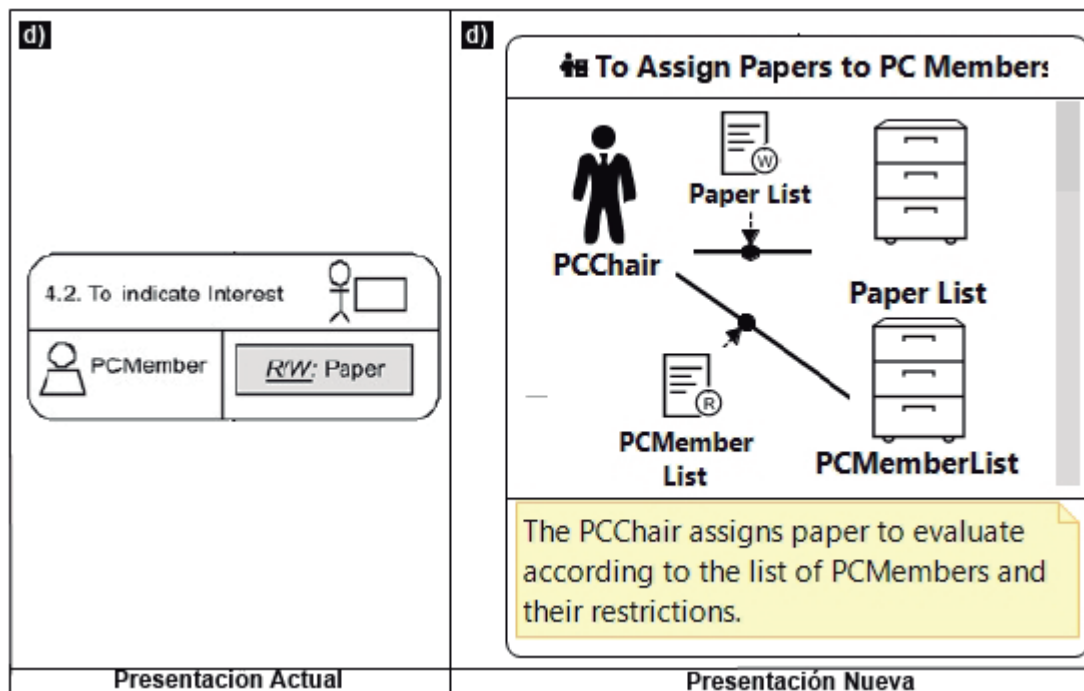



Figura 5.24. Detalle de presentación actual y presentación nueva tarea individual simple d)

Se puede observar que el nuevo aspecto gráfico de las tareas de inter-acción, permite compactar en una sola tarea la representación de un caso de uso de negocio y su respectiva realización. Las tareas interactivas de negocio son las tareas de menor granularidad para representar el grueso del modelado de las tareas, que es el fin de los



diagramas de inter-acción en TD-MBUID. El detalle de estas tareas se muestra a través de las tareas compuestas (cooperativas e individuales), que se descomponen en tareas individuales y tareas de comunicación. El aspecto gráfico de las tareas de comunicación (Figura 5.22), permite por un lado mostrar la participación de los roles y por otro, los mappings directos a los datos y a las interfaces relacionadas con cada tarea. Las modificaciones realizadas a la notación propuesta por CIAF, permiten dar cumplimiento a las **Estrategias 8 y 9**, planteadas en la sección 1.2.

A partir de las tareas individuales y de comunicación se lleva a cabo un proceso de mapping hacia un patrón de presentación a nivel de sistema. Este patrón contiene un conjunto de diagramas, entre otros, el de las tareas interactivas de sistema. En la Figura 5.25 se presenta el aspecto del **Patrón de Presentación** a nivel de sistema. Las tareas de interacción permiten modelar no sólo el mapping existente entre las tareas interactivas y los datos que soportan dichas interacciones, sino también el mapping existente entre dichas tareas y el diálogo y la vista de interfaz concreta. En el diagrama de diálogo se indica el tipo de interacción que se llevará a cabo sobre los datos, si se modifica, se observa o se actualiza un objeto, un atributo o un método. Esta información repercute directamente en los componentes gráficos de la interfaz de usuario que se emplearán para soportar la tarea.

La vista del modelo de datos indica la correspondencia con el modelo dinámico (máquinas de estado), es decir la asociación existente entre las clases de sistema y los diagramas de estado que describen el comportamiento de dichas clases. En el **Anexo D** se encuentra una descripción detallada de todos los elementos de modelado que conforman la notación definida para esta propuesta.

5.3.4 Modificaciones a la herramienta software CIAT

En esta sección se presenta la herramienta denominada AMBUID (*Activity Modeling Based User Interface Design*) una herramienta de software basada en modelos para apoyar a los diseñadores e ingenieros en la creación de modelos basados en la notación utilizada en el marco de desarrollo (**Anexo D**). Esta herramienta permite la edición manual y la generación automática de algunos modelos del marco de desarrollo. AMBUID ha sido desarrollada dentro del marco de modelado de Eclipse que proporciona las herramientas para guiar el modelado de software mediante el uso de conceptos de metamodelado (Moore et al, 2004).

Es importante resaltar que esta herramienta de soporte se ha adaptado a partir de la herramienta de soporte CIAT desarrollada por (Giraldo, 2010), pero siguiendo los lineamientos y mostrando cómo se aplica la definición de la TxA, con lo cual se da cumplimiento a las **Estrategias 10 y 11** planteadas en la Sección 1.2.

Una herramienta de soporte debe contener siempre un equilibrio entre flexibilidad y orientación. Cuanto mayor sea la flexibilidad, el equipo de desarrollo tiene que decidir más por sí mismo cómo utilizar la herramienta. Es por esto que la herramienta tiene que ir acompañada de una metodología. Son las personas que desarrollan software, quienes perciben una mejora de su productividad con el uso de herramientas. Por

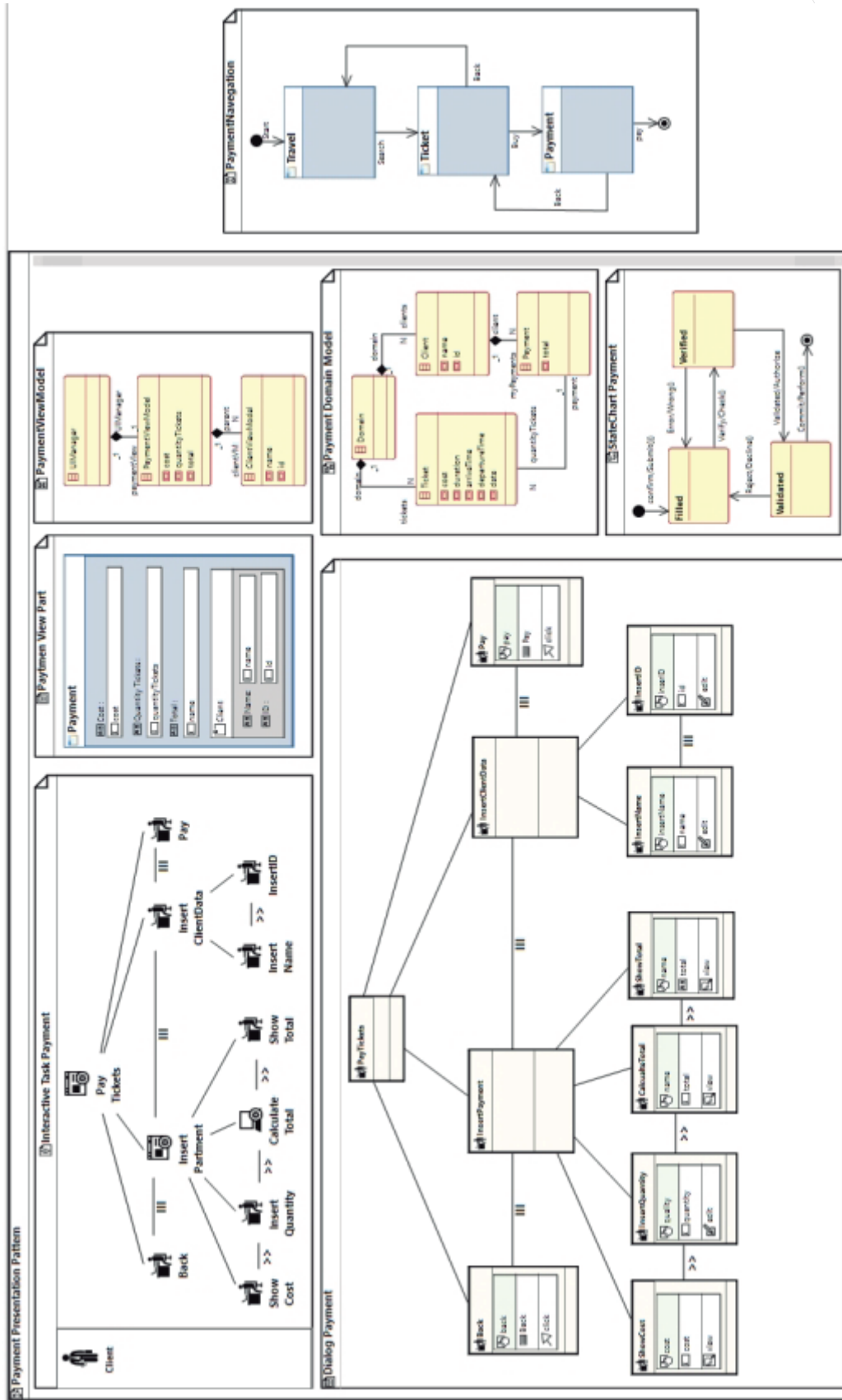


Figura 5.25. Aspecto del patrón de presentación a nivel de sistema.

tanto, se necesitan herramientas para hacer equipos más productivos (Giraldo, 2010).

Estas propiedades de las herramientas pueden ser alcanzadas a partir del uso de entornos de desarrollo que las soporten, por ejemplo: Eclipse. La herramienta AMBUID está implementada como un plugin de Eclipse mediante el uso de EMF (*Eclipse Modeling Framework*) y GMF (*Graphical Editing Framework*).

Eclipse proporciona soporte para MDE, por medio de EMF, durante el desarrollo de modelos a partir de lenguajes. EMF es el marco de modelado de Eclipse y proporciona mecanismos para la persistencia de los modelos en forma de un archivo XML. EMF está diseñado para facilitar el diseño e implementación de modelos de estructurados, porque los conceptos de modelado y sus implementaciones están directamente relacionadas (Budinsky et al, 2003), (Moore et al, 2004).

El Marco de Edición Gráfica (GEF) facilita el desarrollo de las representaciones gráficas de los modelos a partir de representaciones realizadas a través de *Draw2D*, como una adaptación de SWT que es el estándar de dibujo de Eclipse. Mediante la combinación de GEF y EMF, es posible implementar un *plugin* dentro del marco de trabajo de Java para Eclipse. El objetivo de GMF es la definición de herramientas de modelado gráfico en múltiples dominios (Gronback, 2009).

En el desarrollo de AMBUID ha sido necesario involucrar un conjunto amplio de tecnologías. Aunque el desarrollo de los distintos módulos que componen la herramienta final ha sido abordado con distintos intereses y fueron desarrollados de manera independiente, sus ventajas pueden ser combinadas mediante un método específico para producir efectos de sinergia integrada. De esta manera, es posible la configuración de una herramienta de desarrollo que soporta el marco descrito en esta tesis. La Figura 5.26 ilustra dichas tecnologías y su forma de uso.

La complejidad del desarrollo de se considera alta por interactuar en diversos campos de la informática. Cada tecnología ha sido aplicada en una etapa particular para soportar un aspecto específico de AMBUID. Inicialmente se decidió la estructura del marco de trabajo tecnológico a partir de la combinación de todos los *plugins* para que sean utilizados por medio de sus propias vistas y perspectivas. Posteriormente, se define la estrategia mediante la cual es posible interrelacionar todos los modelos para que computacionalmente puedan ser procesados al estar entre ellos relacionados.

La definición de la notación para modelar los diferentes diagramas que soporta la herramienta, ha sido a partir de la notación mejorada de CIAN, como se muestra en CIAT (Giraldo, 2010), es decir, ha sido necesario actualizar esta notación para poder procesar dichos modelos mediante las herramientas de Eclipse. Para facilitar el desarrollo de la herramienta, se especificó la sintaxis concreta de la notación dentro de un solo metamodelo.

La especificación de un solo metamodelo no implica que se genere una sola sintaxis dentro de la herramienta. Solo implica que están empaquetados en una misma descripción EMF. Por lo tanto, los diagramas de cada nivel de abstracción (negocio y sistema), son procesados por medio de plugins independientes. Sin embargo, en esta

herramienta, es suficiente un solo fichero XMI (*modelo.tooldataform*) para describir un modelo completo de un proceso de negocio para un sistema interactivo.

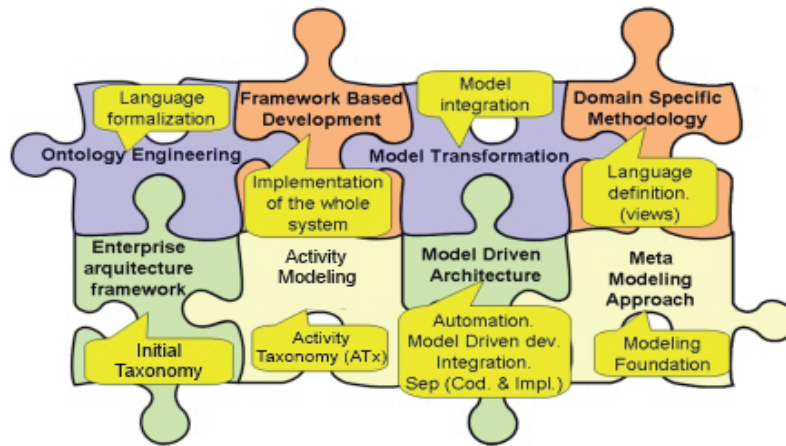


Figura 5.26. Tecnologías utilizadas en el desarrollo de la herramienta, adaptada de (Giraldo, 2010)

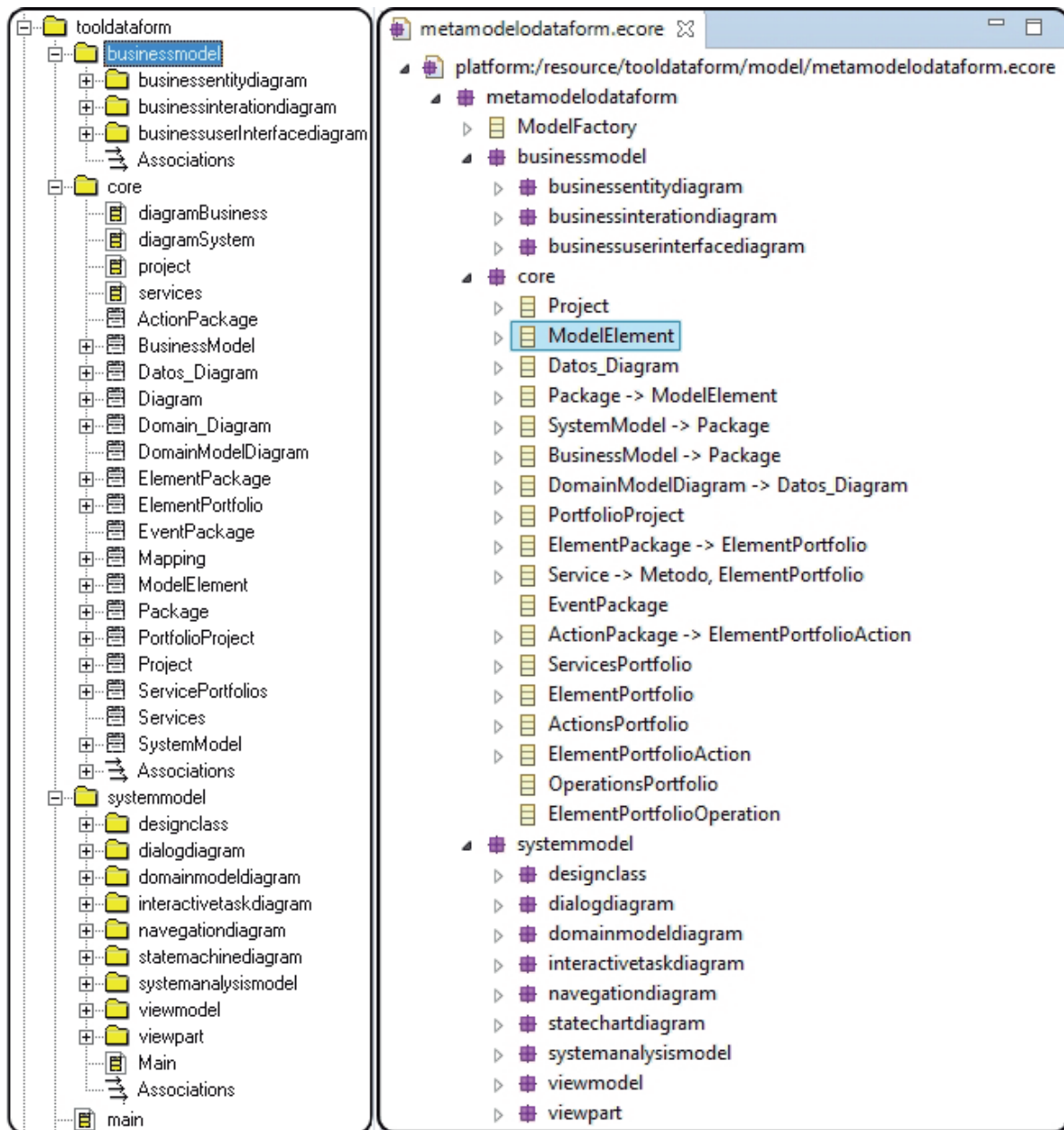
En este sentido, puede verse el metamodelo final como un gran lenguaje de representación de sistemas interactivos, por poseer su propia sintaxis abstracta y concreta que está en capacidad de describir los distintos aspectos del sistema. En la Figura 5.27 se ilustra cómo está organizado el metamodelo de AMBUID dentro del entorno de desarrollo de Eclipse. Por motivos de claridad en la explicación de la herramienta, no se han incluido en este capítulo los distintos diagramas del metamodelo de AMBUID, estos están organizados en **Anexo E**.

5.3.4.1 Desarrollo de la Herramienta

El *Graphical Modeling Framework* (GMF) permite combinar la funcionalidad de los marcos de trabajo EMF y GEF. Es importante aclarar que la herramienta AMBUID está compuesta por dos editores de modelado gráfico: "*business.diagram*" y "*system.diagram*". Para el desarrollo de estos dos editores se siguió la estructura de un proyecto basado en GMF y en EMF. La Figura 5.28 ilustra un diagrama de bloques de la estructura de un proyecto GMF; en este caso se presenta la estructura de los archivos creados para la implementación del editor gráfico a nivel de negocio "*business.diagram*" de la herramienta de soporte AMBUID. Una meta de GMF es permitir que la definición gráfica de una herramienta de modelado pueda ser reutilizada en varios dominios. Esto se logra al usar un modelo de mapping separado para enlazar las definiciones de la herramienta y el modelo específico de dominio (Gronback et al, 2006).

El desarrollo de una herramienta por medio de un proyecto GMF se lleva a cabo mediante los siguientes pasos: Definición del modelo de dominio, Creación de archivo generador de modelos, Definición del modelo gráfico (sintaxis concreta), Generación de la paleta de herramientas, Definición del modelo de mapping, Creación del generador del plugin y Generador de código. Con el desarrollo de la herramienta se ejecutan las **Estrategias 12 y 13** planteadas en la Sección 1.2.

Como se puede observar (Figura 5.27.b), existe un paquete principal denominado “metamodelodataform” que contiene los demás paquetes que conforman la notación. Esta decisión de diseño se toma porque el lenguaje inicia la instanciación de sus elementos de



a Metamodelo de AMBUID **b** Paquetes de AMBUID (meamodelodataform.ecore)

Figura 5.27. Diagramas y paquetes del metamodelo de AMBUID (metamodelodataform.ecore).

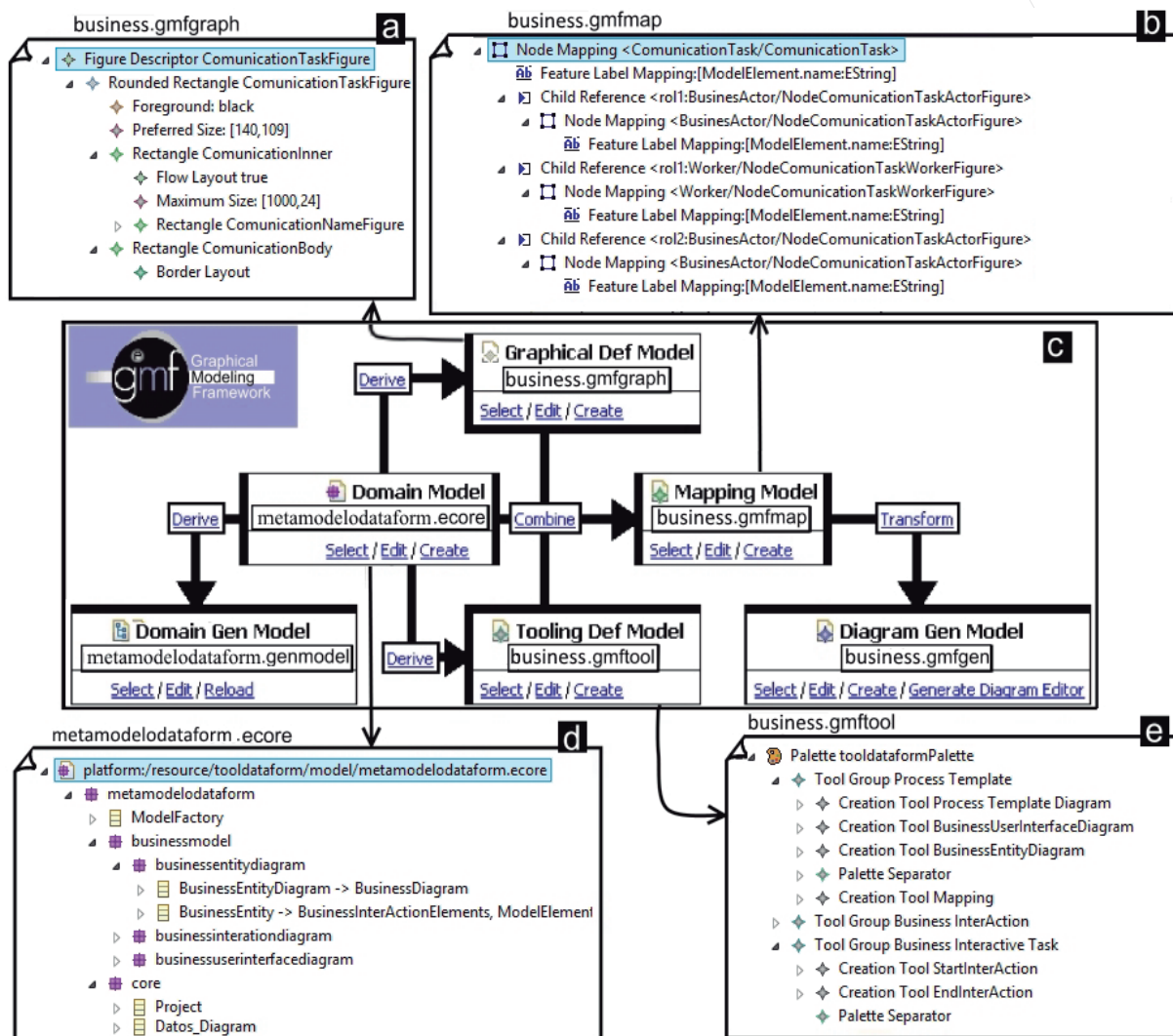


Figura 5.28. Estructura de un proyecto GMF, caso de la herramienta AMBUID.

5.3.4.2 Definición del modelo de dominio

El modelo de dominio dentro de un proyecto GMF es una representación EMF del metamodelo que define la sintaxis abstracta del lenguaje que se desarrolla. Esta sintaxis abstracta actúa como una guía para la comunicación de todos los artefactos y herramientas que intervienen en el desarrollo de una herramienta de soporte a MDE (Giraldo, 2010).

Una técnica común para la especificación de lenguajes es definir primero la sintaxis del lenguaje y luego describir su semántica estática y dinámica (OMG, 2008b). La sintaxis define qué elementos de modelado existen en el lenguaje y cómo estos elementos de modelado se construyen en términos de otros. Cuando el lenguaje tiene una sintaxis gráfica, es importante definir la sintaxis en una notación de forma independiente (es decir, para definir la sintaxis abstracta del lenguaje). La sintaxis concreta se define

entonces mediante un *mapping* entre la notación y la sintaxis abstracta.

La especificación llevada a cabo en esta tesis es el resultado de combinar un lenguaje base de especificación (EMOF), un lenguaje de restricciones sobre objetos (OCL), y el lenguaje natural.

El desarrollo del modelo de dominio dentro de Eclipse se hace mediante un modelo *Ecore* en EMF. Esto permite refinar la estructura y la semántica usando el lenguaje de restricciones a objetos (OCL), y soportar diversas transacciones, consultas y validación sobre los modelos. Generalmente, los modelos vienen provistos de un lenguaje de definición basado en enfoques tradicionales como el BNF, sin embargo, un modelo que se describe en términos de *Ecore* es más expresivo y puede tener asociado un número de sintaxis concretas definidas para la generación de editores gráficos o de texto (Gronback, 2009).

En la Figura 5.29 se presenta un fragmento de la sintaxis abstracta y concreta para el diagrama de interacción de negocio (business interaction). El diagrama de la izquierda describe la sintaxis abstracta, define los componentes del diagrama y sus relaciones desde el punto de vista conceptual, de manera aislada de la forma como será elaborado dicho diagrama por parte de los usuarios desarrolladores. El diagrama de la derecha representa una fracción de las sintaxis concreta de un *Process Template Diagram*. Es una fracción porque no se muestra como es la relación de estos elementos con respecto a los tipos "ProcessTemplate", "Project", "Diagram", "ModelElement", etc., los cuales forman parte de la gestión de diagramas y modelos de la herramienta.

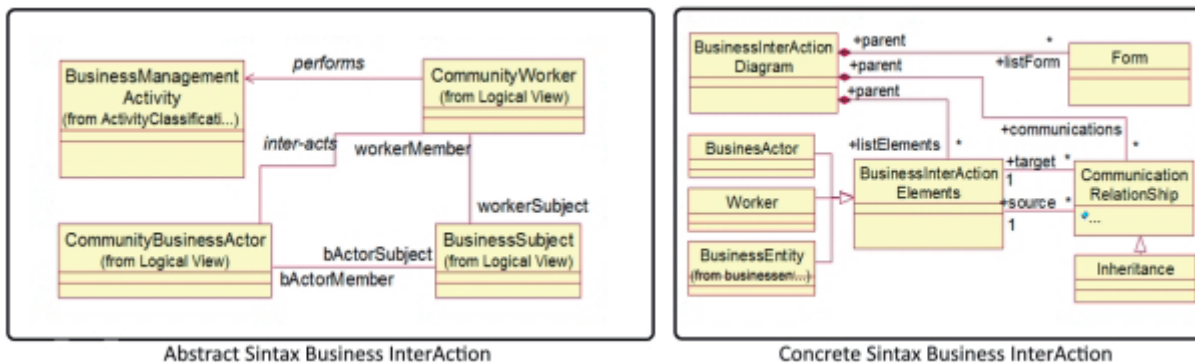


Figura 5.29. Fragmento del Modelo de dominio para la notación en AMBUID.

En GMF una herramienta de modelado se compone básicamente de dos tipos de elementos (nodo y enlace) para representar cualquier elemento de modelado en un lenguaje en particular. Para el caso del *Process Template*, un nodo corresponde a la clase "Business Actor" y un enlace corresponde a la clase "Communication Relationship" de la sintaxis concreta.



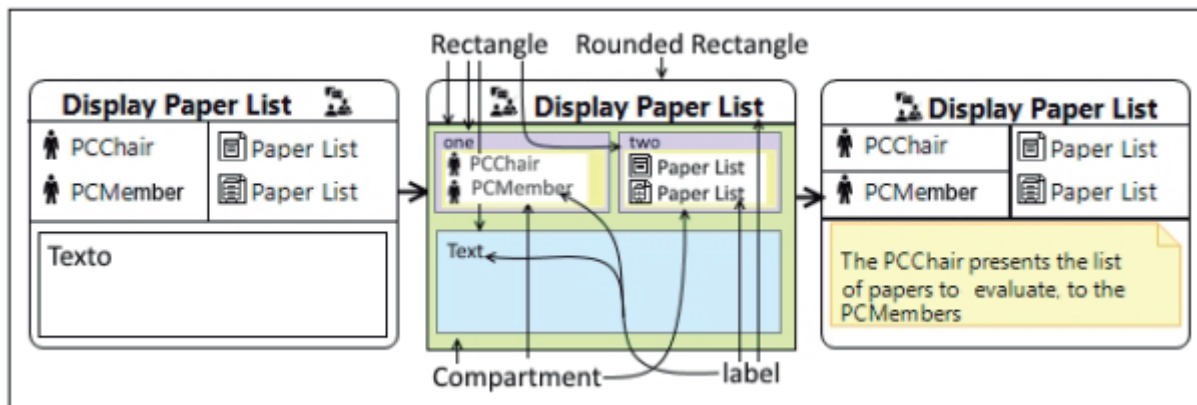
5.3.4.3 Creación del archivo generador de modelos

El generador de modelos EMF es un modelo, que se obtiene a partir del metamodelo Ecore, que es necesario para la generación de las clases del dominio en lenguaje Java. El generador de modelos EMF, tiene una extensión de archivo "metamodelodataform.genmodel" y, es esencialmente un modelo decorado de su respectivo metamodelo Ecore. Este generador de modelos utiliza una serie de plantillas predefinidas que son especificadas en el lenguaje de emisión de plantillas Java (JET). La tecnología JET es utilizada para la transformación de modelos a texto a partir de modelos ECORE.

5.3.4.4 Definición del modelo gráfico (sintaxis concreta)

La especificación de la sintaxis concreta debe ser uno de los pasos preliminares en la especificación de cualquier lenguaje para definir sus particularidades y el *mapping* con respecto de la sintaxis abstracta. Está especificación tiene como partida el análisis visual de cómo se presentarán al usuario desarrollador los elementos de modelado del lenguaje. Este modelo define la estructura de componentes gráficos usados para representar los conceptos.

En la Figura 5.30 izquierda se ilustra el diseño gráfico de una tarea de Comunicación. Este diseño debe, posteriormente, ser especificado a partir de los elementos de modelado visual que dispone GEF para la implementación en GMF (Figura 5.30 centro). Este diseño debe considerar aspectos de *layout*, etiquetado, estructura y contención. Cada componente de la notación gráfica del lenguaje que se diseña debe ser especificado utilizando los elementos del modelo gráfico GEF, por ejemplo: rectángulo, etiqueta, nodo, enlace, etc.



● **Figura 5.30.** Especificación gráfica de la Tarea de Comunicación.

Como en todos los entornos de implementación es necesario posteriormente elaborar la codificación del diseño, en este caso, el diseño visual (Figura 5.31). De esta forma se obtiene el código de la representación gráfica de la herramienta en forma de un archivo de extensión ".gmfgraph" (Figura 5.28a). En la Figura 5.30 derecha se presenta el resultado final de la implementación de la tarea de comunicación. Puede observarse

que su aspecto es bastante similar al análisis y diseño previo.

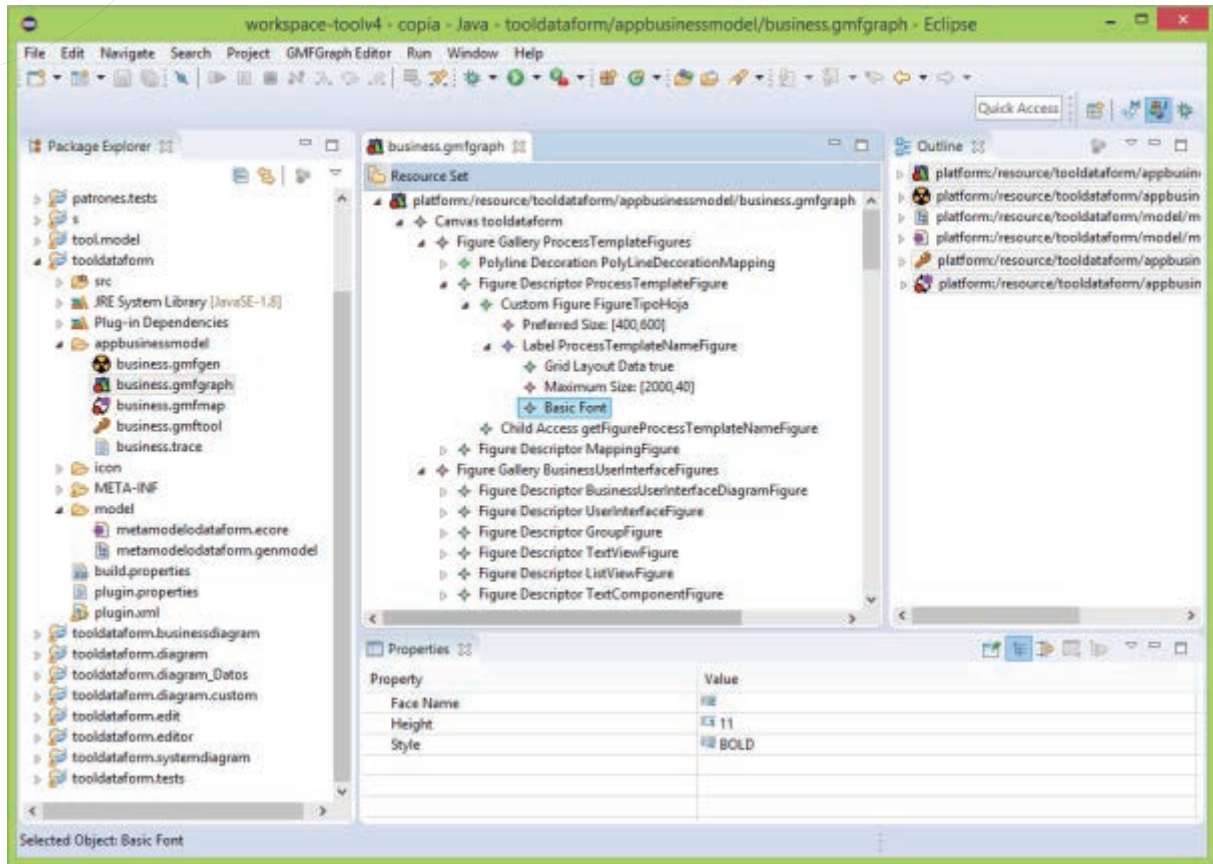


Figura 5.31. Implementación de la notación en GMF.

5.3.4.5 Generación de la paleta de herramientas.

Las herramientas de modelado, usualmente, incluyen paletas de herramientas y otras opciones para manipular los elementos de modelado. El propósito del modelo de paleta de herramientas es especificar estos elementos. La paleta de herramientas está compuesta por la barra de herramientas y diferentes menús que pueden ser definidos para un diagrama (Figura 5.32).

Mediante el uso de GMF es posible especificar todos y cada uno de los componentes de la barra de herramientas. Esta operación es bastante simple y consiste en crear instancias de cada uno de los botones de la barra de herramientas para que sea reconocida por el modelo de *mapping*.

5.3.4.6 Definición del modelo de *mapping*

El modelo de *mapping* es quizás el más importante de todos los modelos de GMF. En éste, los elementos de la definición del diagrama (nodos y enlaces) se asignan al modelo de dominio y se les asigna elementos de la paleta. El modelo de *mapping*

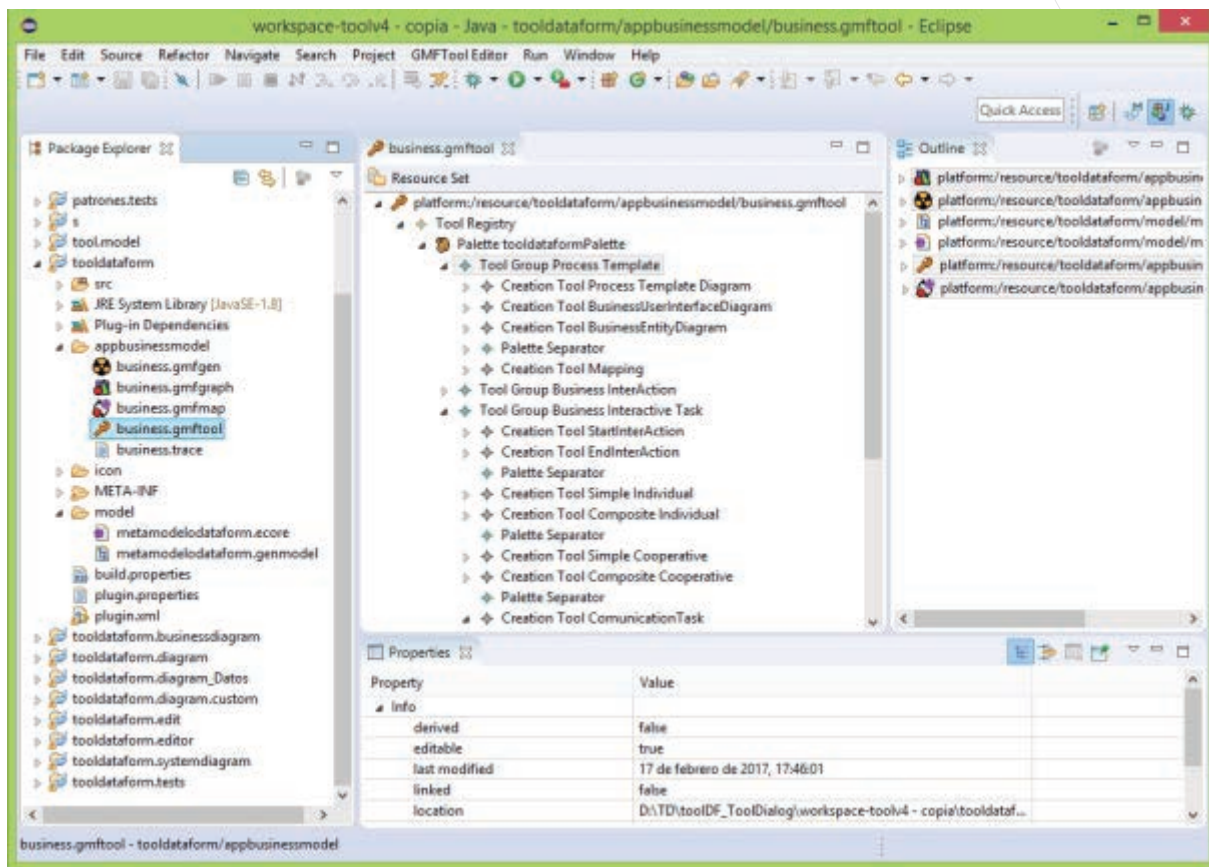


Figura 5.32. Implementación de la paleta de herramientas en GMF.

representa la definición del plugin de diagrama y es utilizado para la creación del modelo de generación de diagrama de extensión ".gmfgen". El diseño de los distintos diagramas da lugar a la elaboración de un modelo de *mapping*. El modelo de *mapping* usa el lenguaje OCL de muchas maneras, por ejemplo: para la inicialización de atributos, definición de restricciones y definición de auditorías y métricas en el modelo.

En la Figura 5.33 se presenta el modelo de *mapping* "business.gmfmap" del nivel de negocio de la herramienta AMBUID. Mediante este modelo es posible especificar todas y cada una de las relaciones de contención de los componentes gráficos para obtener instancias de los objetos del dominio con el objetivo de hacer un *mapping* con su información. Adicionalmente, se relacionan los elementos gráficos entre sí y con la paleta.

Antes de llevar a cabo un modelo de *mapping* es necesario conocer muy detalladamente la estructura de la sintaxis abstracta y concreta. Desafortunadamente, no siempre es posible obtener representaciones gráficas que se adecuen a una estructura deseada de modelo de dominio. Esto se debe a que en GMF, cuando se arrastra un componente gráfico sobre el canvas, se está obligado a crear una instancia de dicho elemento de modelado dentro de la clase de dominio que está representando el *canvas* en ese momento determinado.

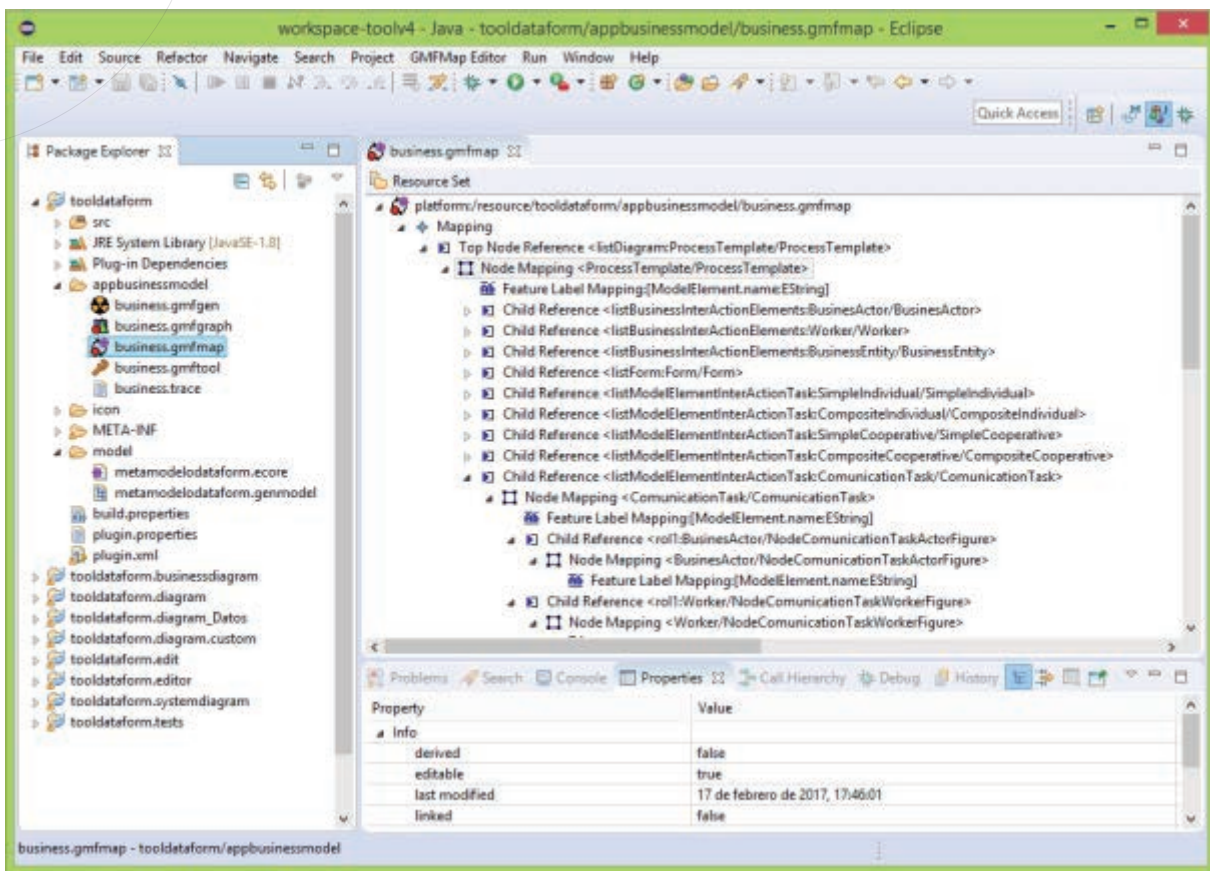


Figura 5.33. Definición del mapping para la herramienta AMBUID.

5.3.4.7 Creación del generador del plugin

AMBUID considera a cada diagrama como una envoltura que abarca un conjunto de elementos de modelado con respecto a un aspecto muy particular de diseño.

Por ejemplo, cuando se hacen los diagramas de diseño de negocio, se describen las entidades de negocio que participan en un proceso de negocio en particular. A efectos de la generación automática de las interfaces de usuario de negocio, interesa conocer cuál es la participación de cada entidad de negocio de manera individual en relación con un proceso concreto.

Cada diagrama permite llevar un registro de la información que es importante dentro del contexto de ese diagrama. La información de cada elemento de modelado que pertenece a un diagrama es cotejada con la información de ese mismo elemento de modelado que está en el nivel de la carpeta de modelos. Así se va enriqueciendo, poco a poco, la especificación de todos los elementos de modelado en el dominio. Cada vez que se agrega un elemento de modelado en un diagrama se analiza la existencia de dicho elemento de modelado dentro del dominio y, así, se verifica la unicidad de los mismos.

Para el control de las transformaciones y, por ende, la generación automática se tiene

un mayor interés sobre los diagramas, en lugar de, abordar el modelo único de elementos de modelado. La organización de diagramas se hace de manera convencional, por medio de paquetes.

Finalmente, todos los diagramas que intervienen en el diseño de la herramienta deben ser elaborados. En la Figura 5.34 se presenta la configuración de los dos principales diagramas que intervienen en la herramienta. El concepto diagrama en GMF no es el mismo que en los lenguajes de modelado. Un diagrama en GMF es una herramienta completa de modelado. Para el caso de AMBUID, se ha creado una herramienta para cada nivel de abstracción expresado en el mapa de ruta de esta propuesta (Negocio y Sistema).

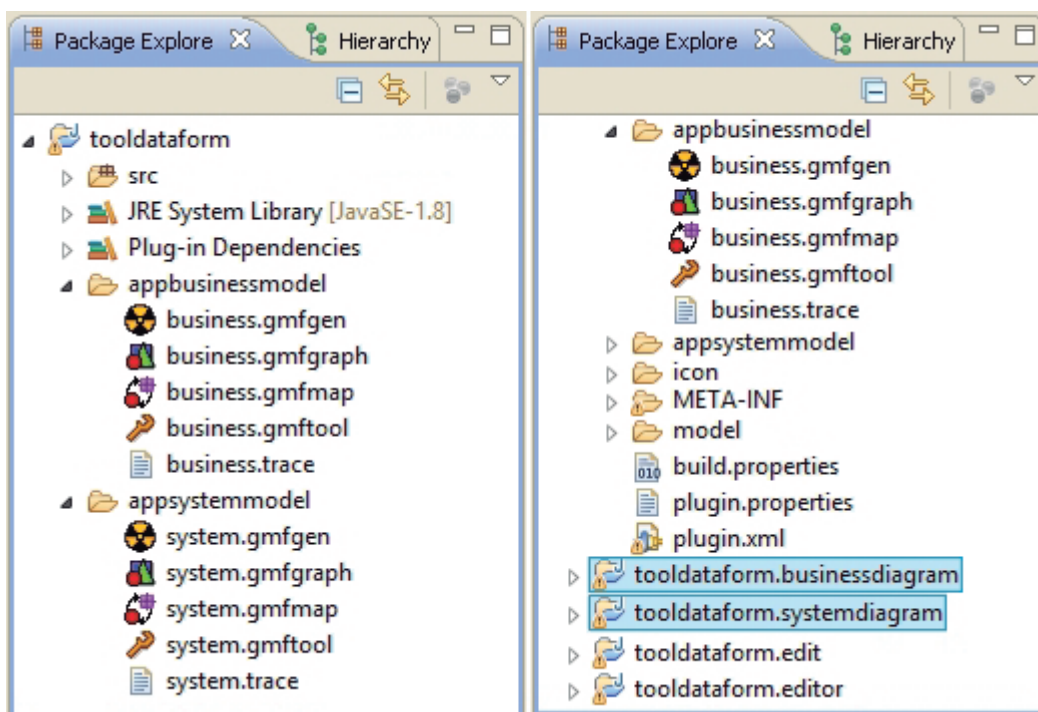


Figura 5.34. Modelos GMF para los dos principales diagramas de AMBUID.

5.3.4.8 Generación de código

El lenguaje utilizado para la implementación de las transformaciones: modelo a modelo y modelo a texto, es el lenguaje Java. Aunque existen muchos lenguajes para llevar a cabo estas transformaciones tal como el lenguaje ATLAS Transformation Language (ATL) y aunque se reconoce su utilidad para este tipo de implementaciones, se decide usar java porque aporta más flexibilidad para este caso.

A partir de esta descripción detallada del marco de desarrollo, los desarrolladores pueden llevar a cabo su proyecto. En la siguiente sección se presenta un caso de estudio mediante el cual es posible explorar otros escenarios de aplicación.

5.3.5 Caso de estudio

En esta sección se presenta la aplicación del marco de desarrollo propuesto a un caso de estudio. Se desarrolla el caso de estudio de "compra de un ticket de avión". Ha sido elegido por ser un ejemplo en el que se dan ciertas necesidades de interacción.

5.3.5.1 Enunciado del caso de estudio

El objetivo principal del proceso o subsistema a desarrollar es soportar la compra de un ticket de avión a través de una agencia de viajes.

Actores del proceso

A continuación se definen las principales responsabilidades que tienen asignadas los actores del sistema:

- *Agente de viajes:* toma contacto directo con el cliente cuando le ofrece y define un viaje. Este rol es el que recepciona, asesora al cliente, concreta la venta de los servicios turísticos, emite la documentación de viaje para cliente, y, se ocupa de la facturación y cobro.
- *Cliente:* solicita asesoramiento con respecto al viaje y a la compra de tickets, decide las características que tendrá el viaje y realiza el pago de los tickets de viaje.

Proceso a desarrollar

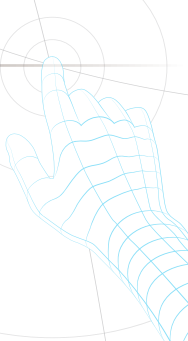
- *Compra de tickets de avión:* El Cliente solicita al Agente de Viajes información y asesoría sobre el viaje que desea realizar. El Agente de Viajes brinda la información necesaria al Cliente y de acuerdo a dicha información el Cliente decide el destino del viaje, el tipo, la cantidad de pasajeros y procede a realizar la compra de los tickets aéreos.

5.3.5.2 Desarrollo del proceso siguiendo el roadmap propuesto

En esta sección se explica cómo se siguen los pasos en el desarrollo de la interfaz de usuario de acuerdo al mapa de ruta propuesto en este trabajo, presentado en la sección 5.3.1.

Modelado del negocio

Se define el proceso a desarrollar y a partir de esta definición se define la Plantilla de Proceso con sus diagramas de Interacción de Negocio y de Tareas de Negocio. Se definen también los Diagramas de Interfaces de Usuario de Negocio y los Diagramas de Entidades de Negocio. Estos artefactos serían equivalentes a una realización de Caso de Uso de Negocio.



En este caso, se trabaja con la actividad de mayor nivel de granularidad, *Compra de tiquetes de avión*.

La definición de la Plantilla de Proceso inicia con la identificación de las posibles intenciones de los usuarios potenciales del futuro sistema. Estas intenciones se definen a través de una división y distinción entre las actividades que son llevadas a cabo por el usuario, durante la realización de un proceso de negocio, y las acciones que puedan ser automatizadas, es decir las acciones realizadas por el Worker. En el nivel de Tareas Interactivas de Negocio, se detallan estas tareas a través de la especificación de una serie de pasos o acciones de negocio. Dichas acciones estarán representadas por medio de modelos de tareas interactivas en el nivel de sistema. En un principio, en la especificación, no se sabe si una acción se llevará a cabo con apoyo de la tecnología o será una actividad manual.

En la Figura 5.35 se muestra la Plantilla de Proceso "*Compra de tiquetes*". En esta plantilla se describen las interacciones de negocio (Figura 5.35a), parte estructural del proceso y las actividades de negocio que llevan a cabo el *Agente de Viajes* y el *Cliente* para describir la dinámica de dicho proceso (Figura 5.35b). El análisis de tareas interactivas de negocio da lugar a una descomposición de tareas.

En el diagrama de interacción de negocio (Figura 5.35a), cada relación entre actores y entidades representa una o más actividades del negocio. Las entidades de negocio que soportan el proceso se representan mediante el Diagrama de Entidades de Negocio (Figura 5.36). Estas entidades representan un modelo inicial del contexto para el diseño de la interfaz de usuario de negocio, representada a través del Diagrama de Interfaz de Usuario de Negocio (Figura 5.37).

El Diagrama de Interfaz de Usuario de Negocio se crea a partir de los datos, entidades que son manipulados dentro del contexto de los procesos de negocio. El objetivo es diseñar un número de Diagramas de Interfaces de Usuario de Negocio que tenga concordancia con el número de sub-tareas de negocio. Se busca que dichas interfaces de negocio puedan soportar un conjunto de tareas de menor granularidad y no a cada una de ellas de manera aislada. Se destaca que las Interfaces a este nivel muestran los datos pero no los botones, menús u otras funciones. Lo que se busca con esta técnica es tener un justo equilibrio entre dos extremos: el diseño de la interfaz de usuario basado en los datos y el diseño de las interfaces de usuario basado en las tareas. Estas interfaces deben ser revisadas y refinadas junto con los usuarios, hasta que lleguen a representar los modelos mentales que éstos tienen para cada tarea del proceso a desarrollar. La estructura de cada Interfaz dependerá, en gran medida, del modelo mental que tiene el usuario acerca de la información que manipula en las actividades.

La Figura 5.37 presenta una primera versión de la Interfaz de Usuario para el proceso "*Comprar Tiquetes de Avión*". Aún no se ha obtenido un modelo completo de la interfaz, ya que sus componentes deben ser refinados con la información que se obtiene a partir de las tareas y de los requerimientos del usuario. Todo esto se realiza con la ayuda de herramientas de modelado y transformación de modelos.

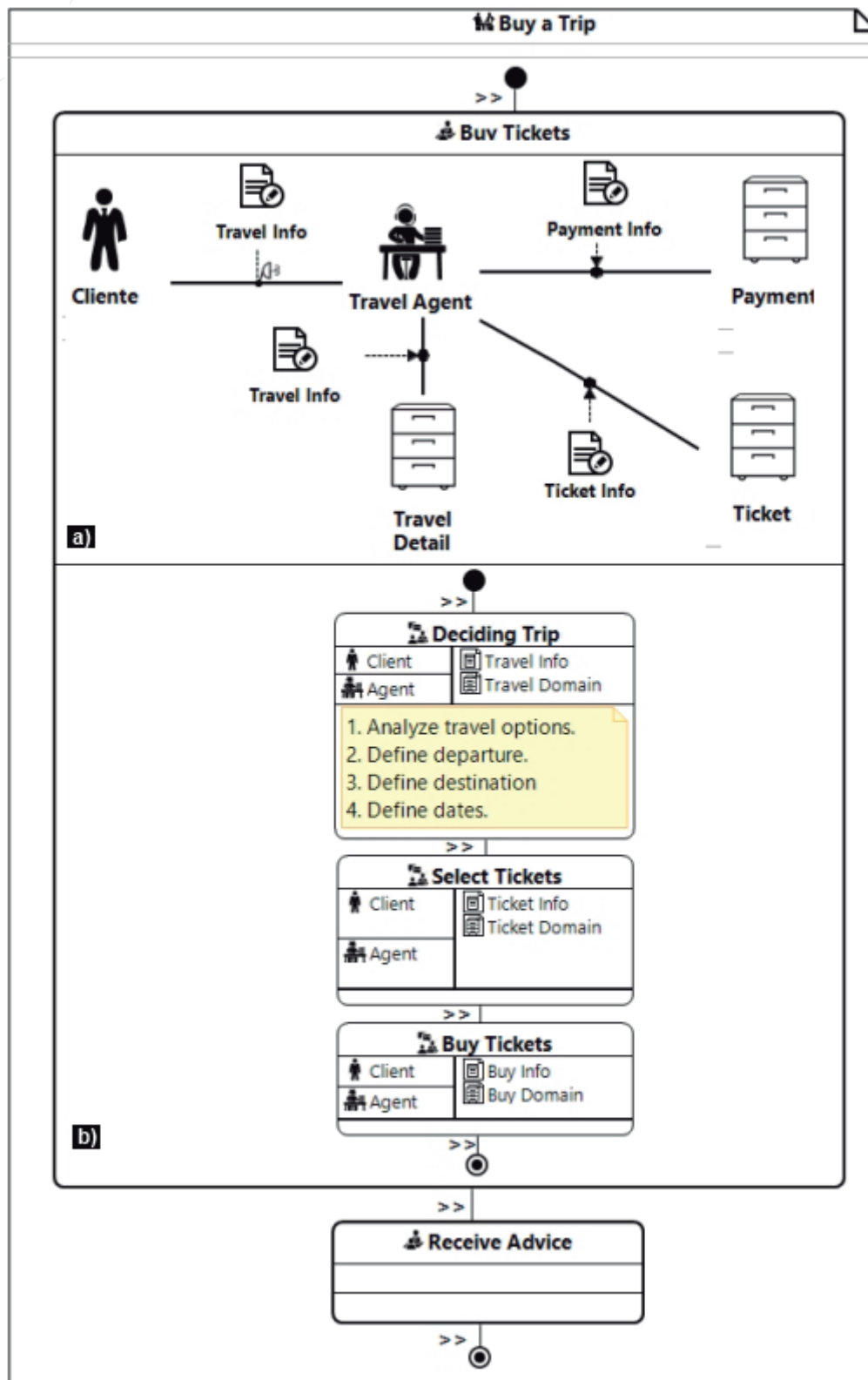


Figura 5.35. Plantilla de Proceso para "Compra de Tiquetes"

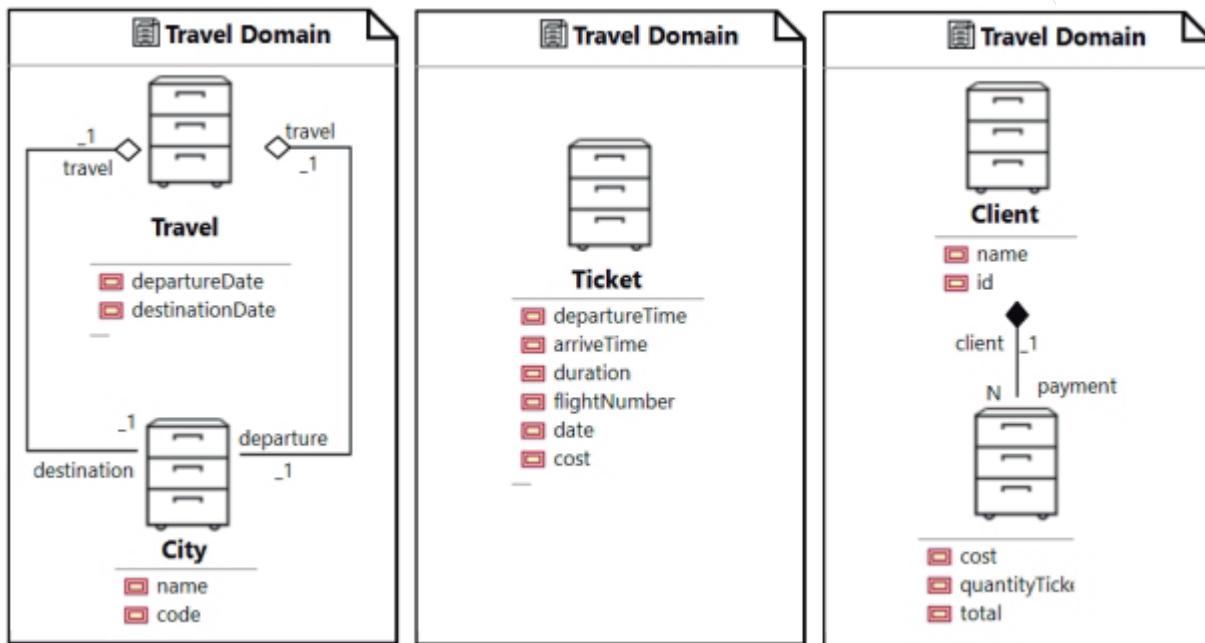


Figura 5.36. Diagrama de Entidades de Negocio

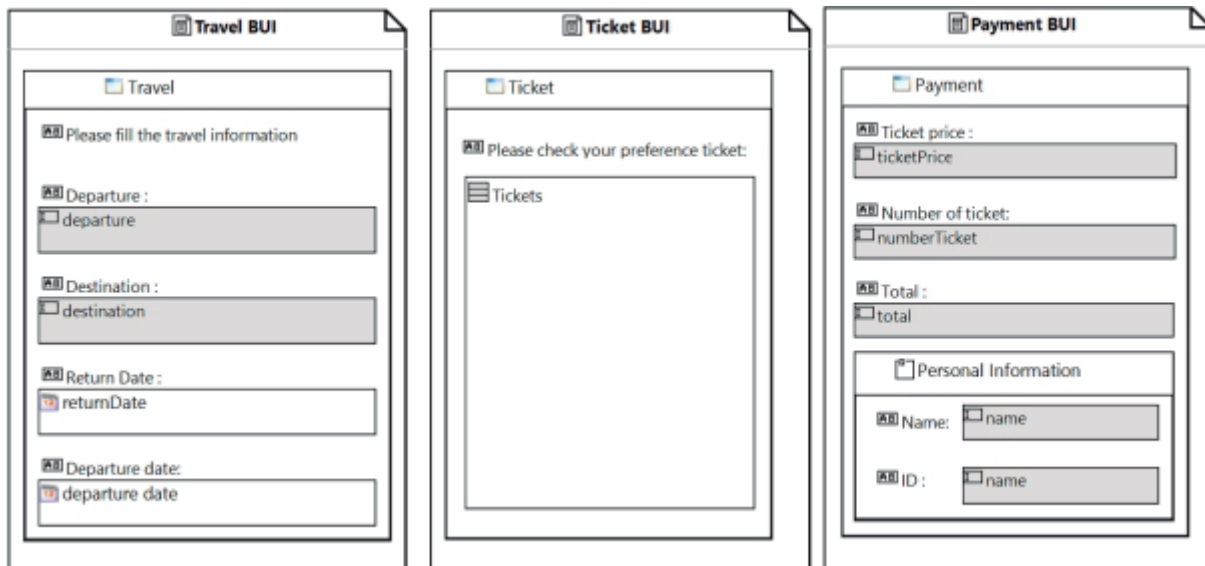


Figura 5.37. Diagrama de Interfaz de Usuario de Negocio

La Figura 5.38 muestra que en el nivel de modelado de negocio por cada tarea interactiva de negocio, se construye un Diagrama de Entidades de Negocio y un Diagrama de Interfaz de Usuario de Negocio. Es decir, existe un mapping entre cada tarea y cada pareja de diagramas, de entidades y de interfaces de usuario (Figura 5.38).

A partir del conjunto de diagramas definidos en el nivel de abstracción de negocio, es posible continuar con el diseño del proceso en el nivel de abstracción de sistema.

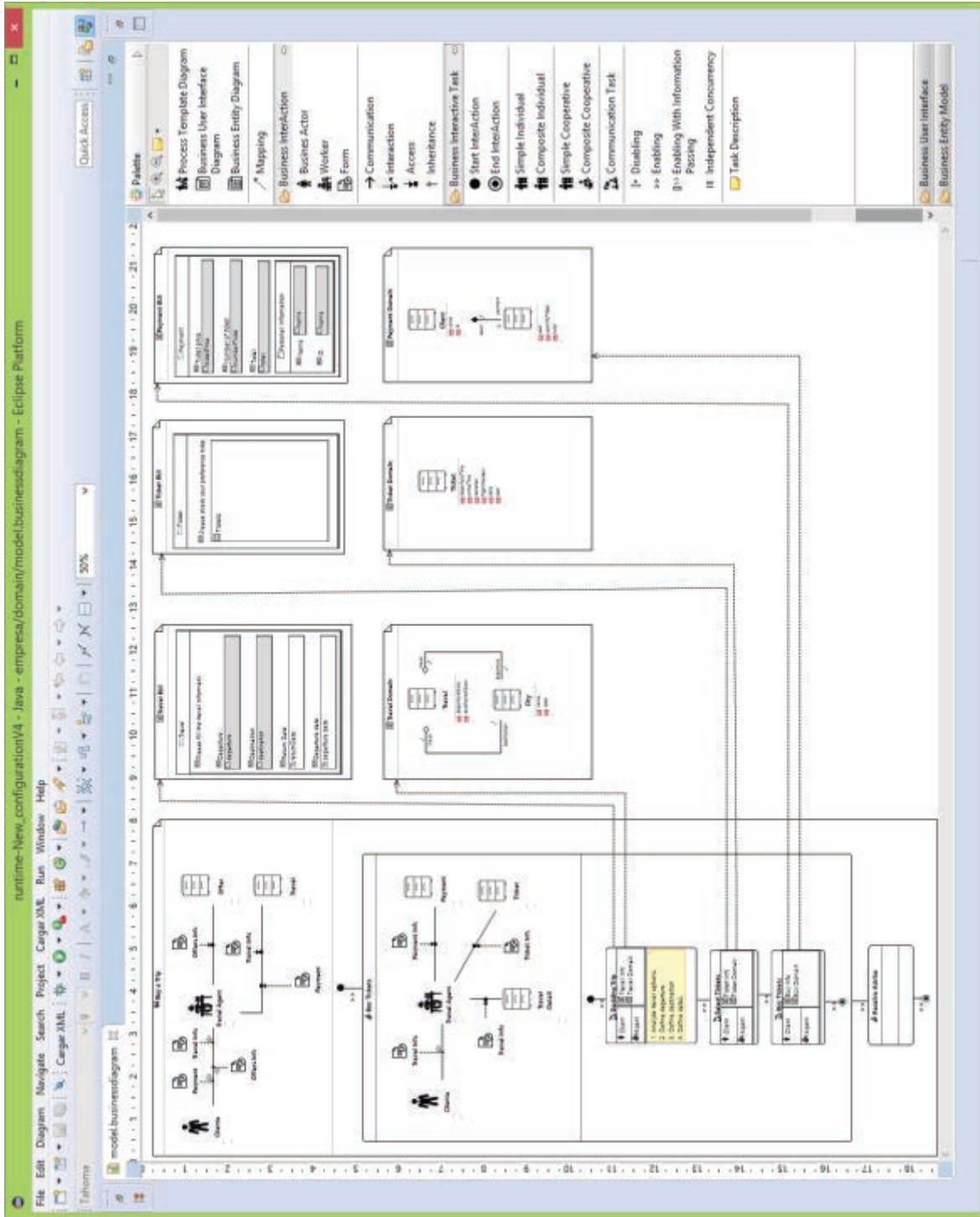


Figura 5.38. Modelado de negocio para el proceso “Comprar Tiquetes de Avión”.



Modelado del sistema interactivo

Una vez definida la Plantilla de Proceso y sus Diagramas de Entidades y de Interfaces asociados, es posible definir el Patrón de Presentación y el Diagrama de Navegación a nivel de Sistema. El Patrón de Presentación contiene un conjunto de diagramas que permiten definir las interacciones entre el usuario y el sistema interactivo, y su relación con los datos, ciclos de vida, navegación, diálogo, e interfaz de usuario. Los diagramas son los definidos en la sección 5.3.2: **1. Diagrama de Tareas Interactivas** (*Interactive Task Diagram*), **2. Diagrama de Diálogo** (*Dialog Diagram*), **3. Vista de la Interfaz Concreta** (*View Part Diagram*), **4. Vista del Modelo de Datos** (*View Model Diagram*), **5. Modelo de Dominio** (*Domain Model Diagram*), **6. Diagramas de Estados** (*StateChart Diagram*). Para cada una de las tareas especificadas en la Plantilla de Proceso, es preciso crear un Patrón de Presentación (Figura 5.39).

El propósito de la Figura 5.39 es mostrar la distribución de los diferentes tipos de diagramas dentro del patrón de presentación, para la tarea interactiva de negocio "Pagar Tiquetes". Por otro lado, las Figuras 5.40 a 5.44 muestran en detalle cada diagrama que conforma el patrón de presentación para este ejemplo y algunos de los mapping entre ellos.

La Figura 5.40a muestra el diagrama de tareas interactivas, que indica el usuario responsable de realizar dichas tareas. Normalmente, las tareas que tiene asignadas un Worker a nivel de negocio, son las tareas que se automatizan, es decir, la realización del Worker está representada por el Sistema Interactivo.

Aunque el diagrama de tareas surge a partir de las acciones de las actividades de negocio, es posible, encontrar nuevas tareas si se aplica un recorrido cognitivo (con los usuarios) a partir de las Interfaces de Usuario de Negocio. El usuario define la secuencia de tareas que le permite cumplir con el objetivo definido por el proceso.

A partir de cada diagrama de diálogo (Figura 5.40b) es posible definir la propuesta de Interfaz de Usuario Concreta, o *View Part* (Figura 5.41a). Para este trabajo, dicha interfaz se genera utilizando el algoritmo de Limbourg (Limbourg, 2004) donde la estructura jerárquica de las tareas determina el nivel de agrupación de los elementos de la interfaz. Los tipos de componentes que conforman la *View Part* se determinan a partir de la información de la multiplicidad entre los elementos del *View Model*. Algunos componentes están asociados a acciones, y por tanto, se convierten en botones.

La estructura (*layout*) y apariencia de la interfaz concreta es el resultado de un proceso de edición manual con los usuarios, que se basa en fuentes de conocimiento tanto de diseño de interfaces como del dominio del problema.

Los Diagramas de Estados representan la orquestación de la funcionalidad definida para las clases de sistema (Figura 5.42). Cada clase de sistema puede tener asociada más de un diagrama de estados. En este caso se muestra el ciclo de vida de la clase "Pago", en el que se avanza a medida que ocurren los eventos especificados en su ciclo de vida. De esta forma se está indicando que el desarrollo de la funcionalidad se centra principalmente en los ciclos de vida de las clases de sistema.

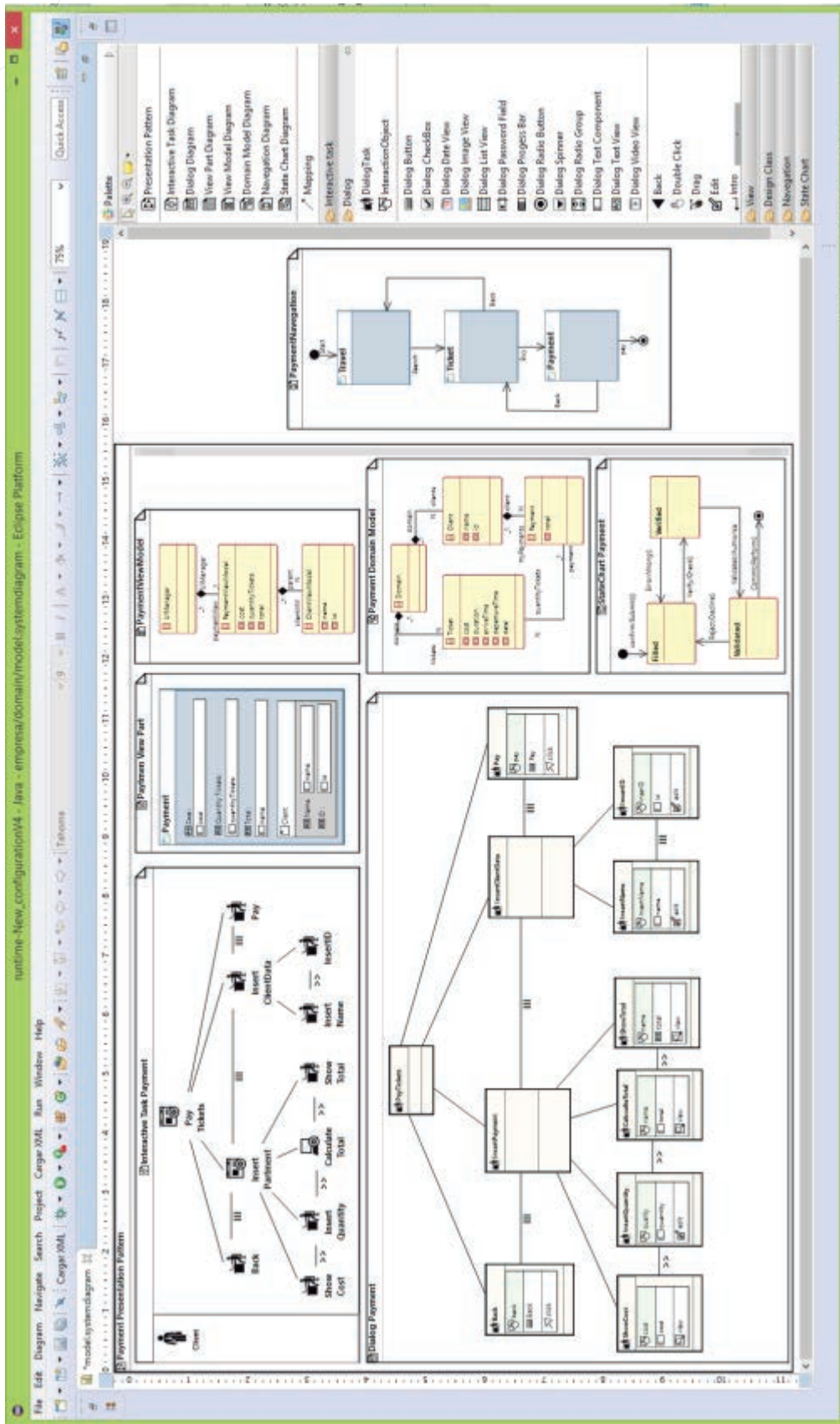


Figura 5.39. Patrón de Presentación y diagrama de navegación para la tarea "Pagar Tiquete de Avión".

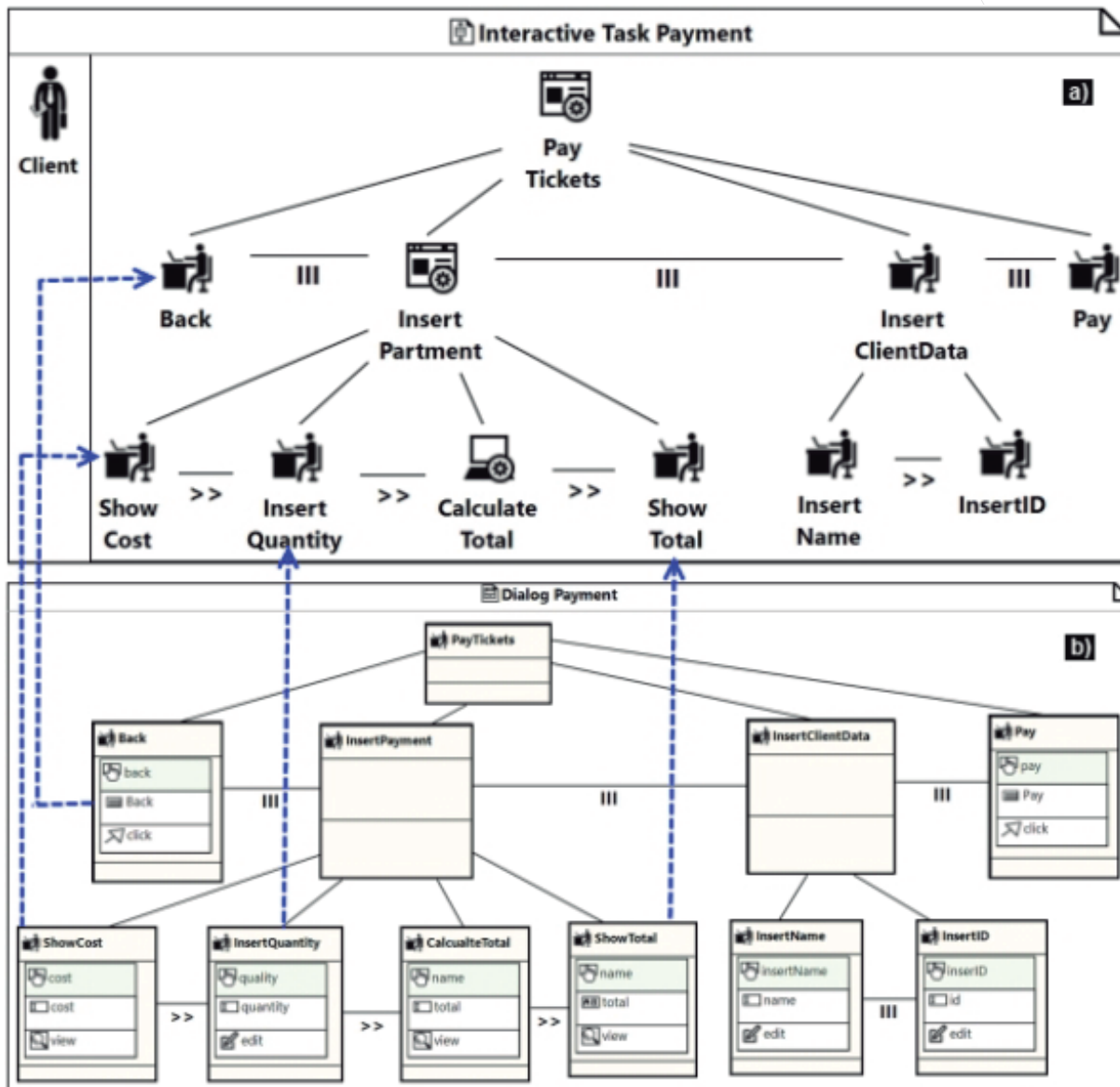


Figura 5.40. Diagrama de tareas interactivas y diagrama de diálogo

En el caso del desarrollo de la interfaz de usuario, los diagramas de estado son útiles para la definición de las interfaces que están relacionadas con el cambio de estado de cada entidad (Giraldo et al. 2009e).

El diagrama de Navegación es el elemento de orquestación entre patrones de presentación, mediante las acciones que permiten conmutar el sistema interactivo desde una tarea hacia las demás tareas interactivas (Figuras 5.43 y 5.44). Sólo se diseña un diagrama de navegación por cada proceso que se desea implementar.

La herramienta software AMBUID brinda una serie de vistas de propiedades que permite implementar los mappings entre los diagramas y sus elementos. Como es el caso de las conexiones entre acciones en la navegación, eventos en las máquinas de

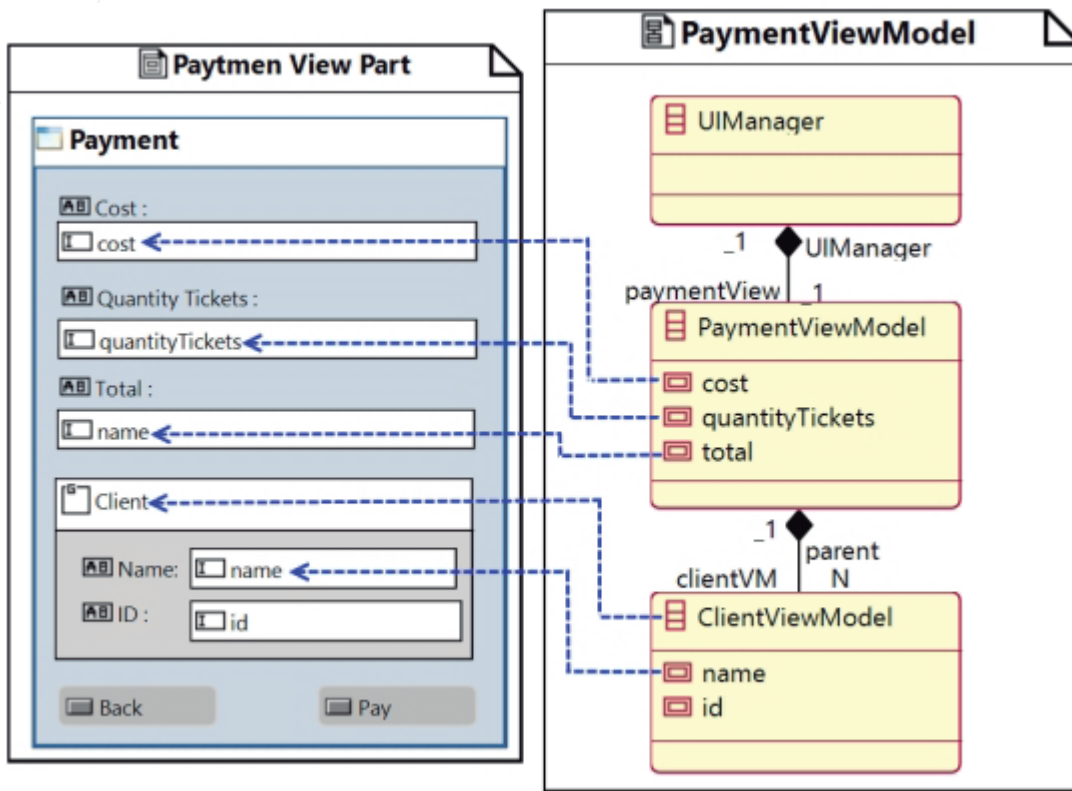


Figura 5.41. Mappings entre View part y view model

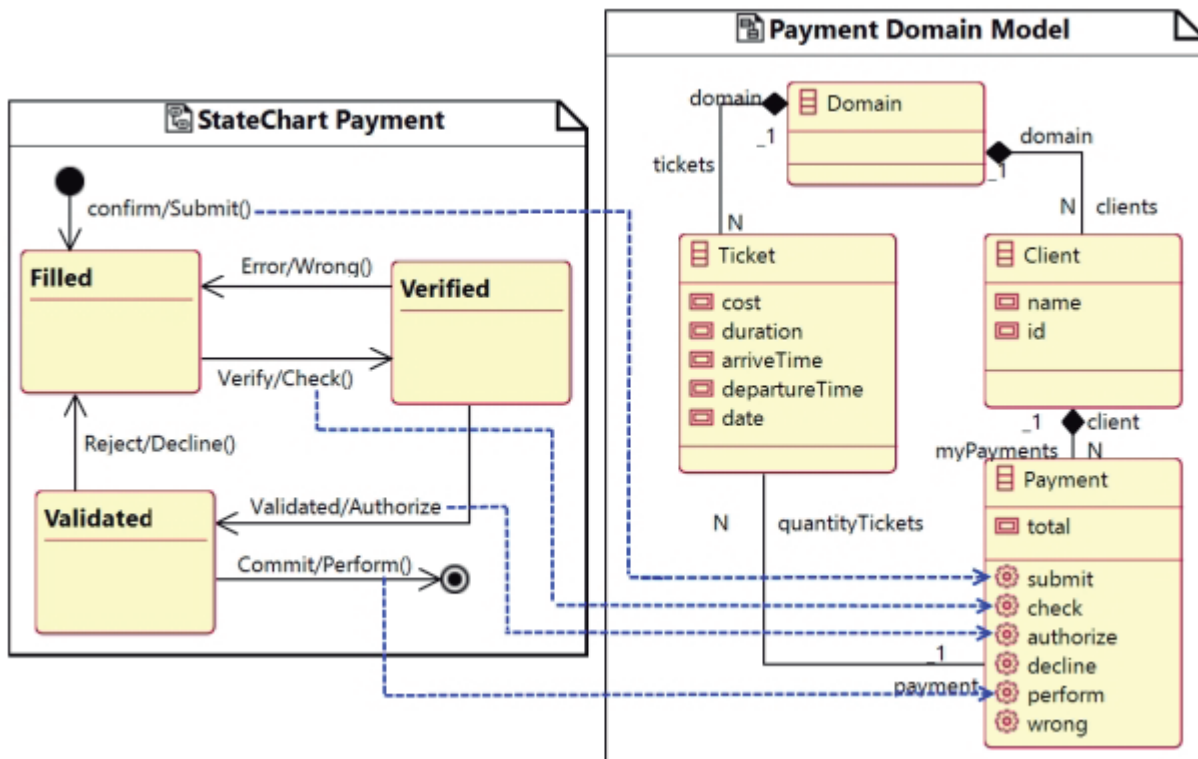


Figura 5.42. Mappings entre diagrama de máquina de estados y diagrama de dominio

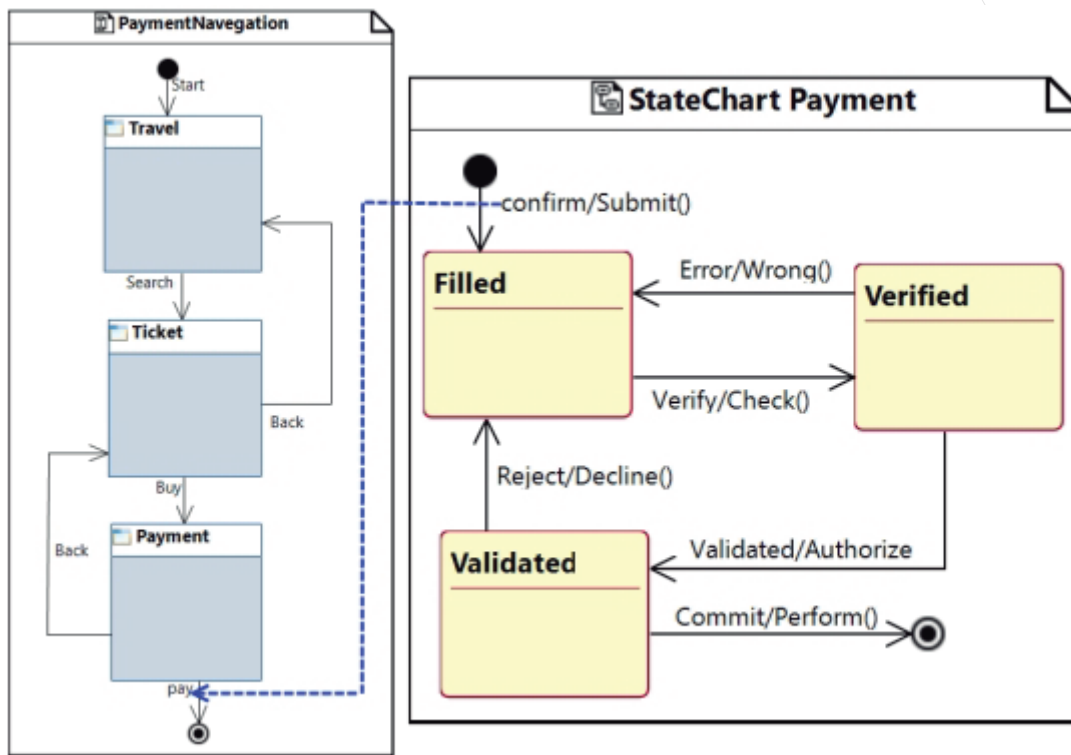


Figura 5.43. Mappings entre diagrama de navegación y máquina de estado

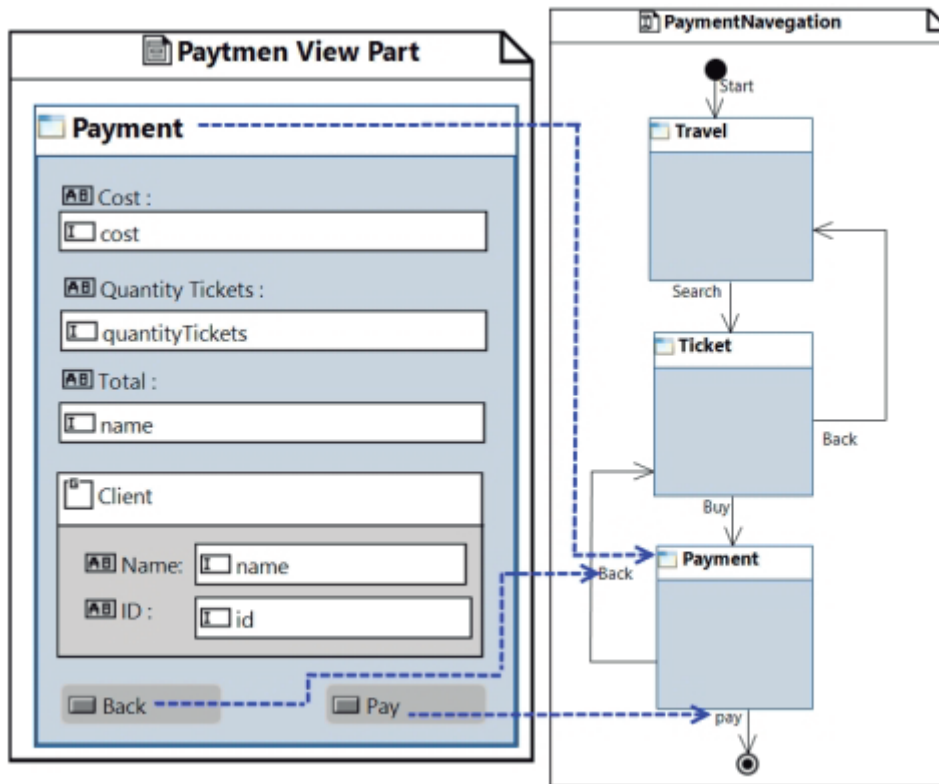


Figura 5.44. Mappings entre diagrama de navegación y view part

estado y las operaciones de las clases de sistema. Esta vista también permite introducir el código (*semantics*) asociado a dichas acciones y operaciones que representan la funcionalidad del sistema interactivo. Las Figuras 5.45 y 5.46 presentan dos de las vistas de propiedades (*acciones y eventos*) asociadas a la funcionalidad de la tarea "Pagar ticket de avión".

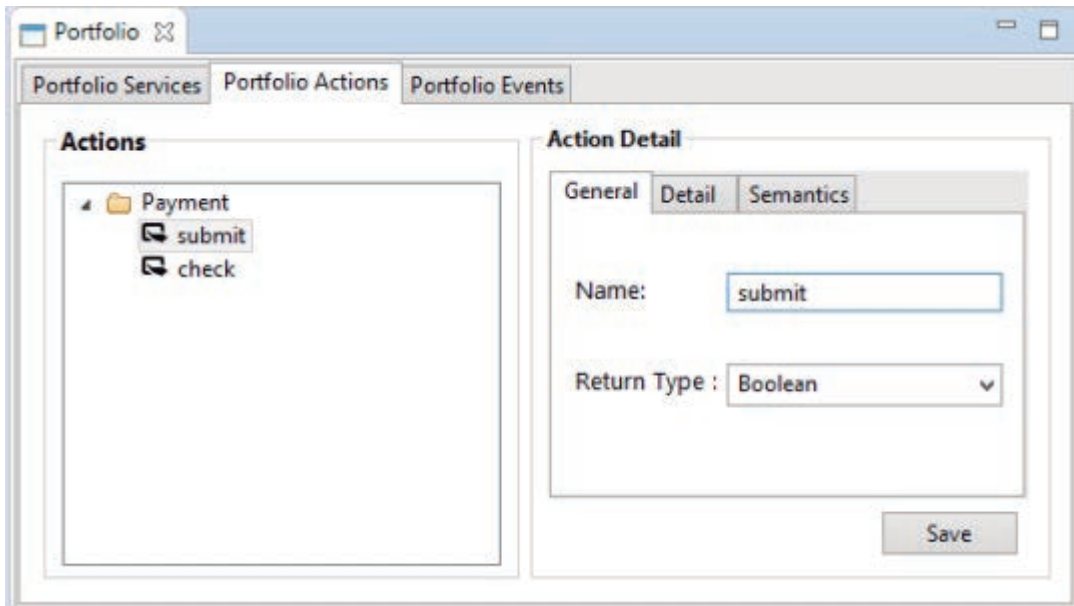


Figura 5.45. Vista de propiedades para las acciones

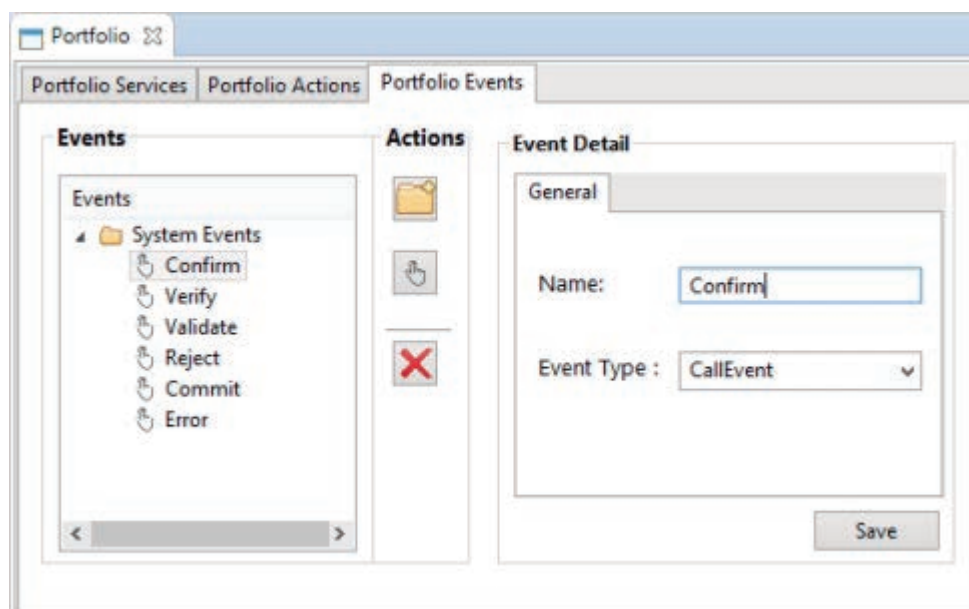
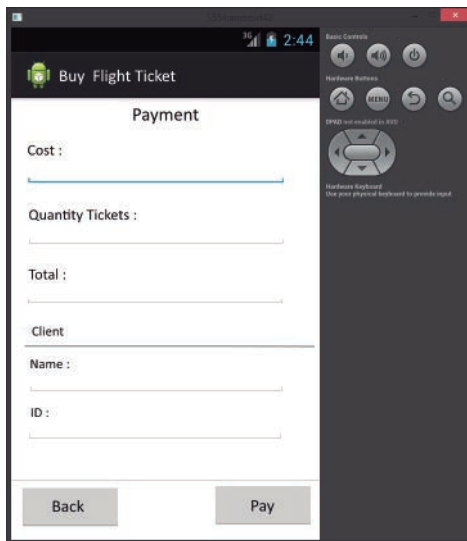


Figura 5.46. Vista de propiedades para los eventos

Una vez que se han adoptado una serie de decisiones sobre el método y la tecnología que se utilizará, se aborda la fase final del desarrollo de la interfaz de usuario. Como paso final de este proceso se llevan a cabo una serie de transformaciones. Estas transformaciones utilizan la información que se ha recopilado a lo largo de todo el proceso sobre las tareas, los datos, los diálogos, la navegación, los ciclos de vida y los elementos de la interfaz.

La Figura 5.45 muestra un ejemplo de la interfaz final para la tarea "Pagar ticket de avión" en Java y XML. Para este caso, la interfaz final se ejecuta sobre el sistema operativo Android.



Estas interfaces finales se generan a partir de transformaciones de modelo a texto que involucran representaciones concretas en cada una de las tecnologías de implementación.

Figura 5.47. Interfaz de usuario final para la tarea "Pagar ticket de avión"

En este capítulo se han presentado los usos propuestos para el Marco Conceptual, Metodológico y Tecnológico definido en esta tesis. De esta forma, se puede decir que se ha realizado una valoración preliminar del valor práctico y la utilidad de este trabajo.



6

CONCLUSIONES

6. CAPÍTULO SEXTO

6. CONCLUSIONES Y TRABAJO FUTURO

6.1 CONTEXTO

6.2 RESUMEN DE LAS CONTRIBUCIONES

6.2.1 Contribuciones teóricas y conceptuales

6.2.2 Contribuciones a nivel metodológico

6.2.3 Contribuciones a nivel tecnológico

6.3 TRABAJO FUTURO

6.4 APORTACIONES PRINCIPALES

6.4.1 Publicaciones

6.4.2 Proyectos de investigación

6.4.3 Pasantías y estancias de investigación



6. CAPÍTULO SEXTO

6. CONCLUSIONES Y TRABAJO FUTURO

6.1 CONTEXTO

Este capítulo resume la contribución realizada por la presente tesis a la Definición de Marcos Conceptuales, Metodológicos y Tecnológicos para el Modelado y la Ejecución de la Actividad de Sistemas Interactivos, con respecto a los siguientes aspectos: 1) Contribuciones teóricas y conceptuales relacionadas con la definición de un Marco Conceptual para HCI. 2) Contribuciones acerca del uso del Marco Conceptual para Evaluación de Artefactos generados en el Desarrollo de Sistemas Interactivos. 3) Contribuciones acerca del uso del Marco Conceptual para la Definición de Marcos de Desarrollo de Sistemas Interactivos.

Según las revisiones realizadas al estado del arte en el campo de marcos conceptuales y taxonomías en Ingeniería de Software y HCI, se destaca que la mayoría de taxonomías propuestas, se definen para un dominio muy específico. Además de que se utilizan para comparar, identificar ventajas y carencias relacionadas con las características de propuestas que encajan en el conjunto de clasificadores definidos para dichas taxonomías.

Con respecto a estas observaciones, el marco conceptual o la taxonomía de la actividad propuesta en esta tesis, permite a los diseñadores de sistemas interactivos analizar la naturaleza de la actividad desde diferentes vistas y puntos de vista. En este sentido, cada uno de los roles interesados en analizar una especificación estará en capacidad de evaluar diferentes aspectos de dicha especificación, por ejemplo, la organización, HCI, comunicación, seguridad, colaboración, etc. Por ejemplo, un experto en el nivel de la organización se enfoca en analizar las interacciones entre las personas, más que analizar las interacciones entre los usuarios y el sistema interactivo. Por ende, la especificación del sistema debe estar clasificada en diferentes niveles de abstracción, negocio y sistema separadamente.

En este sentido, un experto a nivel del sistema interactivo debería enfocarse en analizar las tareas interactivas (Base y soporte), separadamente de sistema (administrativas). Por ejemplo, la notación CTT (Paternò, 2004) hace esta distinción entre tipos de tareas, aunque no realiza alguna definición o distinción de tipos de tareas en el nivel de abstracción de negocio. Lo cual no sería muy atractivo para un experto que se desenvuelve en este nivel.

El marco conceptual se evalúa con un conjunto de validaciones empíricas y algunos análisis para estimar los resultados obtenidos con respecto al problema planteado.

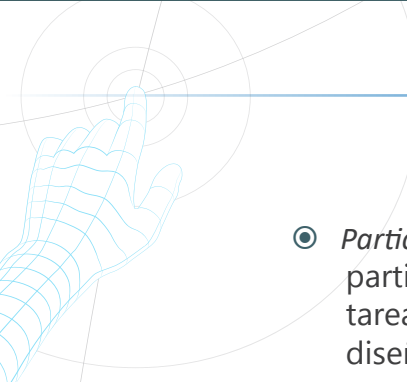
Finalmente, el desarrollo de los casos de estudio para cada uso del marco conceptual propuesto, conduce a un conjunto de resultados que contribuyen al mejoramiento de la propuesta y a identificar los trabajos futuros.

6.2 RESUMEN DE LAS CONTRIBUCIONES

Las contribuciones de esta tesis se resumen según los aspectos que componen la definición del Marco Conceptual, Metodológico y Tecnológico propuesto.

6.2.1 Contribuciones teóricas y conceptuales

- ⦿ *Taxonomía de la Actividad Extendida*: El hecho de llevar un proceso formal de integración entre propuestas, en este caso la Teoría de la Actividad, la Clasificación del Modelado de la Actividad y la Taxonomía de Zachman, provee un soporte teórico en la definición de un metamodelo que describa los elementos relacionados con la Actividad y en la definición en sí de la Taxonomía de la Actividad para Sistemas Interactivos. Se destaca la identificación de la Función (Actividad) como la pieza de orquestación entre las demás columnas que integran la especificación completa de un sistema interactivo. También la identificación del clasificador "inter-acción" en el nivel de abstracción de negocio.
- ⦿ *Taxonomía de la actividad como marco de evaluación*: La Taxonomía de la Actividad propuesta permite evaluar las capacidades de las propuestas que se enfocan en el desarrollo de sistemas interactivos, tanto a nivel de sus notaciones como a nivel de los artefactos de desarrollo propuestos. Básicamente se evalúa el cubrimiento que tiene cada propuesta con respecto a los clasificadores definidos por la Taxonomía.
- ⦿ *Relaciones de mapping entre diagramas que modelan la actividad*: Se identificaron las dependencias existentes entre elementos conceptuales, tanto en el mismo nivel de abstracción, como entre niveles de abstracción. Dichas relaciones se llevan a nivel concreto, entre elementos de modelado. Como es el caso de la definición de mappings existentes entre las "Tareas de Interacción de Negocio" y los "Diagramas de Interfaces de Usuario de Negocio" y "Diagramas de Entidades de Negocio". La identificación y definición de estos mappings contribuyen al diseño de transformaciones para la generación automática de código ejecutable para sistemas interactivos.
- ⦿ *Tarea de comunicación*: Normalmente, los diagramas de tareas presentan el concepto de "Tarea Interactiva", que denota la interacción del usuario con el sistema, sin embargo, en esta tesis se introduce el concepto de "Tarea de comunicación", que permite la especificación de una tarea interactiva a nivel de la organización, es decir la comunicación entre personas. Es importante aclarar que en esta tesis se provee la guía metodológica que indica la trazabilidad de una Tarea de Comunicación con las Tareas que se especifican a nivel del sistema interactivo.

- 
- *Particionamiento de la Interfaz de Usuario:* Se introduce el concepto de particionamiento de la interfaz de usuario a partir de la descomposición de las tareas de interacción a nivel de negocio. Este particionamiento simplifica el diseño de los diagramas de tareas interactivas, de los diagramas de navegación y de diálogo de las interfaces de usuario.

6.2.2 Contribuciones a nivel metodológico

- *Mapa de ruta para el desarrollo de la interfaz de usuario:* Se especifica el mapa de ruta para el desarrollo de la IU basado en el modelado de la actividad en los sistemas interactivos. El mapa de ruta muestra la trazabilidad existente entre los niveles de abstracción Negocio y Sistema, en lo que tiene que ver con la especificación de la Labor de los Sistemas Interactivos. Lo anterior, debido a que HCI se ha enfocado en estudiar la parte la Interacción a nivel del Sistema Informático, pero no en la Interacción a nivel de la Organización, la interacción entre las personas. Se especifican los elementos Rol-Actividad-Artefacto, para cada nivel de abstracción. Se destacan los artefactos "Diagrama de Interfaz de Usuario de Negocio", "Diagrama de Interacción de Negocio", "Diagrama de Tareas Interactivas de Negocio", "Diagrama de Diálogo", y el uso del patrón "View-Model-View". Los elementos que conforman el mapa de ruta propuesto permiten expresar la temporalidad, la sincronización, la jerarquía, y la estructura, como aspectos principales que caracterizan los sistemas interactivos.
- *Modificaciones propuestas a la especificación de casos de uso:* Comprender las carencias y excesos de usabilidad (HCI) en especificaciones de Casos de Uso implica proveer contexto que proviene de las interacciones entre humanos y que posteriormente dirigen las interacciones humano computador. La Estructura Taxonómica de la Actividad propuesta permite identificar los niveles de abstracción y granularidad utilizados en las especificaciones de los Casos de Uso evaluados. Igualmente, permite tener una vista clara de cómo el Modelado de la Actividad está asociado directamente al modelado de la responsabilidad. Es decir, que existe una relación directa entre los roles y las tareas que realizan. La labor siempre es una sola, lo que se muta es la responsabilidad. Adicionalmente, la forma en la que se describe la actividad a través de la Estructura Taxonómica propuesta, permite el desplazamiento, a la derecha o a la izquierda, de las tareas que realizan los actores, de negocio y de sistema, para que puedan ser asignadas a uno, entre ellos, según como se realicen los workers.
- *Notación para clasificadores de la actividad a nivel de de negocio:* La necesidad de facilitar el entendimiento y la manipulación de elementos de modelado empleados en las herramientas software, condujo a la introducción de una representación gráfica asociada para algunos de los clasificadores definidos en la TxA. El desarrollo de lenguajes es muy importante desde la perspectiva del modelado en ingeniería de software porque la implementación de un software que sea funcional, requiere que se capturen con precisión y de forma concisa

todas las propiedades esenciales del sistema de interés. Los ingenieros de software utilizan lenguajes de modelado con el fin de comprender las características relevantes de los sistemas complejos existentes o deseados. Igualmente, para comunicar su entendimiento e intenciones de diseño hacia dichos sistemas. En este sentido, los componentes de un lenguaje de modelado, como su notación abstracta, notación concreta, deben ser acordes a los modelos mentales de los ingenieros, de tal manera que sea posible capturar la información que se pretenda utilizar para bien sea generar una aplicación software o generar un lenguaje que pueda ejecutar la mayoría de tipos de tareas, tipos de funcionalidades definidas en el ámbito de los sistemas interactivos.

6.2.3 Contribuciones a nivel tecnológico

- *Herramientas software para edición de diagramas:* Las herramientas de modelado permiten la edición de los diagramas utilizados en el marco de desarrollo propuesto a partir de la Taxonomía de la Actividad. Estas herramientas procesan la sintaxis concreta de para la notación definida por medio de archivos XMI.
- *Herramienta runtime para ejecución de modelos dinámicos:* Se ha implementado un componente que interpreta los diagramas de máquinas de estados y diagramas de navegación, que permiten ejecutar el comportamiento asociado a estos elementos de modelado.

6.3 TRABAJO FUTURO

El trabajo que se ha llevado a cabo en esta tesis puede servir de punto de partida para iniciar nuevos trabajos de investigación. A continuación, se presentan algunas ideas en esta línea de trabajo:

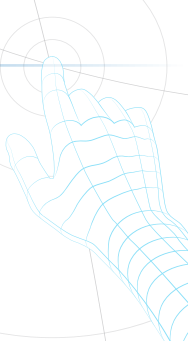
- *Evaluación de propuestas metodológicas para sistemas interactivos:* Las evaluaciones realizadas en esta tesis permiten concluir que tanto en el nivel de negocio como en el nivel de sistema, los aspectos relacionados con la Labor se capturan a través de la técnica de Casos de Uso. Anteriormente, se mencionó la necesidad de incorporar aspectos de HCI a este tipo de artefactos. Sería necesario entonces ampliar este estudio a nivel de Artefactos en lugar de Elementos de Modelado, es decir evaluar los artefactos que resultan durante el proceso de desarrollo de software utilizando la TxA, para concluir si sería necesario incluir, incorporar o trasladar algunas técnicas y mecanismos de HCI desde el nivel del sistema interactivo, hacia el nivel de Negocio. El estudio podría incluir también una evaluación que sea basada en los Diagramas de Lenguajes de Modelado, en lugar de los Elementos de Modelado. Esto porque el diseño de los sistemas interactivos se ve afectado por variables que deben medirse a partir de los Diagramas de los lenguajes de modelado, como Punto Focal, Continuidad, Balance, Equilibrio. Estas

variables hacen que un diseño sea más expresivo, fácil de entender y de comunicar para los ingenieros.

- *Herramientas software para soporte a la clasificación de la actividad:* Una herramienta software que facilite la edición de la especificación en el modelado de la actividad para ubicar cada fragmento de dicha especificación de acuerdo a la estructura taxonómica de la actividad. Por ejemplo, que permita el desplazamiento a la derecha o a la izquierda, de las tareas que realizan los actores, de negocio y de sistema, para que puedan ser asignadas a uno, entre ellos, según como se realicen los workers.
- *Generación automática de diagramas:* Componentes software integrados a la herramienta de soporte para generar de forma automática diagramas para el modelado de la actividad, desde el nivel de abstracción de negocio, hasta el nivel de abstracción de sistema. Por ejemplo, generar automáticamente versiones iniciales de los diagramas de tareas interactivas, a partir de los diagramas de interacción de negocio, y así mismo, los diagramas de diálogo a partir de los diagramas de tareas interactivas. Igualmente, sería posible generar los diagramas de navegación a partir de los diagramas de tareas interactivas de negocio.
- *Mejoras a la herramienta software:* Es necesario realizar mejoras a los editores desarrollados, en cuanto a la presentación y distribución de los modelos que soporta. Es un trabajo que requiere de experiencia por parte de los desarrolladores en los entornos de implementación para herramientas de modelado.
- *Navegador de proceso:* Un navegador basado en web que facilite la navegación y aprendizaje del marco de desarrollo. Este navegador debe proveer múltiples vistas para que cada componente del marco pueda ser estudiado de forma separada. Adicionalmente, que por medio de este navegador, se pueda re-utilizar el contenido de métodos del marco de desarrollo para la definición de otras estructuras de proceso o mapas de ruta.
- *Refinamiento del marco conceptual, metodológico y tecnológico:* Cualquier propuesta es susceptible de mejoras y se espera que esta propuesta pueda ser utilizada en casos reales que permitan contribuir a su refinamiento y actualización.

6.4 APORTACIONES PRINCIPALES

Como producto de esta tesis doctoral, los trabajos y aportaciones preliminares han sido publicados en proceedings de varios congresos y revistas nacionales e internacionales. Se incluyen además otros como proyectos, pasantías y estancias de investigación, que han contribuido a la consecución de los objetivos de este trabajo. A continuación se detallan las contribuciones.



6.4.1 Publicaciones

Revistas

- **María L. Villegas**, William Giraldo, César A. Collazos, Ana Molina, Manuel Ortega. **"Modelo de Integración de Componentes Metodológicos Dentro del Contexto de CIAF"**. Revista Colombiana de Computación - RCC, Vol 14, No 2 (2013).

Este artículo presenta un modelo de integración de componentes metodológicos dentro del contexto del "Marco de Desarrollo de Sistemas Groupware Interactivos (CIAF: Collaborative Interactive Application Framework)". Como caso de estudio se realiza la integración de dos propuestas enmarcadas en el desarrollo de sistemas interactivos, la primera, TD-MBUID (Task & Data Model Based User Interface Development), enfocada en el desarrollo de la interfaz de usuario y la segunda, MPIu+a (Modelo de Proceso de la Ingeniería de la Usabilidad y la Accesibilidad), en la evaluación de la usabilidad.

- William J. Giraldo, Mónica Lorena Tobón, Fáber D. Giraldo, **María L. Villegas**, Alexandra Guerrero, Mónica Yulieth Cortés, Alexandra Ruiz, César A. Collazos. **"Mecanismo de Incorporación de Técnicas de HCI en el Contexto de Colombia"**. Revista Colombiana de Computación - RCC, Vol 15, No 1 (2014).

En este artículo se presenta un caso práctico de incorporación de HCI en empresas colombianas de desarrollo de software, principalmente en la región del Eje Cafetero Colombiano. Este caso está soportado por un modelo de ecosistema autosostenible que surge del acercamiento de la academia a las empresas de desarrollo de software como respuesta a la necesidad de convertir el conocimiento científico en acciones prácticas que puedan ser realizadas por los ingenieros de desarrollo.

- **Maria L. Villegas**, César A. Collazos, William J. Giraldo, Juan González. **"Activity Theory as a Framework for Activity Taxonomy in HCI"**. IEEE Latinoamérica, Vol 4, No 2. Febrero de 2016.

En este artículo se analiza la Teoría de la Actividad para tomarla como base en la definición de la Taxonomía de la Actividad (TxA). Se muestra el uso de la TxA para: la incorporación de HCI en los procesos de desarrollo de software, la definición de un conjunto de clasificadores mínimos para especificar sistemas interactivos que promueven HCI, la evaluación e integración de lenguajes y/o procesos de desarrollo de software, la especificación de modelos de ejecución de sistemas interactivos.

- **Maria L. Villegas**, César A. Collazos, William J. Giraldo, Juan González, Josefina Guerrero. **"The Activity Taxonomy [ATx] as an Evaluation Framework for Modeling Elements in HCI"**. Romanian Journal of Human - Computer Interaction (RRIOC). Aceptado en febrero de 2017. Pendiente de publicación.

Este artículo se enfoca en presentar el uso de la Taxonomía de la Actividad como framework de evaluación de elementos de modelado utilizados para capturar

especificaciones de la actividad en el contexto de HCI. Se analiza cómo los elementos de modelado evaluados corresponden con las categorías definidas para la TxA y también en qué medida resultan apropiados para representar la actividad en la especificación de sistemas interactivos. Adicionalmente, se analiza el nivel de entendimiento que tienen los ingenieros y estudiantes que practican el desarrollo de sistemas interactivos, con respecto a los elementos de modelado evaluados.

- **Maria L. Villegas**, William J. Giraldo, César A. Collázos, Juan González. **“Towards an Activity Taxonomy (ATx) from Separation of Concerns in Use Case Modeling”**. Revista de Ingenierías Universidad de Medellín. Pendiente de publicación. Aceptado en abril de 2016.
 En este trabajo nos proponemos la separación de elementos de negocio y técnicos, al momento de elaborar Casos de Uso, usando la TxA. Presentamos la TxA en el Modelado con Casos de Uso para reducir su complejidad y principalmente para incrementar su efectividad y utilidad en lo que tiene que ver con la Interacción Humano Computador.

Proceedings

- **Maria L. Villegas**, William J. Giraldo, César A. Collazos, Ana I. Molina y Manuel Ortega. **“Methodological Approach for the Languages and Processes Integration within the CIAF Context”**. 8CCC. VIII Congreso Colombiano de Computación. Octubre de 2013.
 Este artículo presenta la ponencia que corresponde con artículo publicado en la revista RCC, 2013.
- **Maria L. Villegas**, William J. Giraldo, César A. Collazos, Toni Granollers. **“Software process implementation method with eclipse process framework composer: MPiu+a case”**. 8CCC. VIII Congreso Colombiano de Computación. Octubre de 2013.
 Este artículo contiene los resultados obtenidos luego de la formalización aplicada al proceso “Modelo de Proceso de la Ingeniería de la Usabilidad y la Accesibilidad (MPIu+a)” con el lenguaje SPEM y su implementación con el Eclipse Process Framework Composer (EPFC).
- William J. Giraldo, Maria L. Villegas, César A. Collazos. **“Incorporation of HCI: Clasification of Activity Modeling”**. 9CCC. IX Congreso Colombiano de Computación. Septiembre de 2014.
 En este artículo se presenta el modelo de clasificación de la actividad se muestra su utilidad marco de referencia para promover la calidad del software principalmente lo relacionado con la Interacción Humano Computador.
- **Maria L. Villegas**, William J. Giraldo, César A. Collazos. **“Activity Taxonomy-Analysis of Proposals for Development of Interactive Systems”**. XV Congreso Internacional de Interacción Persona Ordenador. Interacción 2014. Tenerife, España.

Se presenta un análisis de propuestas metodológicas para el desarrollo de sistemas interactivos, utilizando la clasificación del modelado de la actividad.

- ⊙ William J. Giraldo, Mónica Lorena Tobón, Fáber D. Giraldo, **María L. Villegas**, Alexandra Guerrero, Mónica Yulieth Cortés, Alexandra Ruiz, César A. Collazos. **"HCI Incorporation: a case for Colombia"**. XV Congreso Internacional de Interacción Persona Ordenador. Interacción 2014. Tenerife, España.
Esta es la ponencia que corresponde con el paper publicado en la revista RCC.
- ⊙ William J. Giraldo, Manuel A. Pineda, Robinson Arias, **Maria L. Villegas**, Faber D. Giraldo. **"Herramienta para la generación de código Android a partir de modelos conceptuales"**. XVIII Ibero-American Conference on Software Engineering. Lima, Perú. Abril de 2015.
En este artículo se presenta un prototipo de herramienta para generación de aplicaciones Android, a partir de un conjunto de modelos conceptuales y diagramas. Esta herramienta está concebida para generar el 100% de la estructura funcional de una aplicación Android.
- ⊙ **Maria L. Villegas**, William J. Giraldo, César A. Collazos, Juan González. **"Incorporation of HCI: Usability Validation in Use Cases through the Activity Taxonomy"**. 10CCC. X Congreso Colombiano de Computación. 2015.
En esta propuesta se presenta un método para validar la usabilidad en un sistema interactivo a partir de la especificación de casos de uso. El método propuesto parte de la taxonomía de clasificación de la actividad que pretende soportar la incorporación de la Interacción Humano Computador (HCI) en los procesos de desarrollo de software. Para validar el método se escoge la especificación de un caso de uso para un sistema interactivo transaccional. La discusión de los resultados obtenidos se orienta a las mejoras que proporciona el método propuesto en términos de las variables relacionadas con la interacción y en la potencialidad que tiene para mejorar la incorporación de HCI desde los casos de uso por medio de la taxonomía de la actividad.
- ⊙ **Maria L. Villegas**, William J. Giraldo, César A. Collazos, Juan González. **"Towards an Activity Taxonomy (ATx) from Separation of Concerns in Use Case Modeling"**. SICC 2015. Seminario Internacional en Ciencias de la Computación. Universidad de Medellín, Colombia, Octubre de 2015.
Esta es la ponencia que corresponde con el paper aceptado para publicación en la Revista de Ingenierías de la Unviersidad de Medellín.
- ⊙ **Maria L.Villegas**, William J. Giraldo, César A. Collazos, Juan González. **"Evaluación de Elementos de Modelado en el Desarrollo de Sistemas Interactivos"**. 11CCC. XI Congreso Colombiano de Computación. 2016.
Esta es la ponencia que corresponde con el paper aceptado para publicación en la Revista Romanian Journal of Human - Computer Interaction (RRIOC).

- **Maria L.Villegas**, William J. Giraldo, César A. Collázos, Juan González, Josefina Guerrero. **"Activity Taxonomy: Selecting Modeling Elements for Interactive Systems"**. Interacción 2016. XVII International Conference on Human Computer Interaction. Salamanca, España. Septiembre de 2016.

En este artículo se muestran los resultados obtenidos en el proceso de evaluar un conjunto de elementos de modelado para la especificación de Sistemas Interactivos, a partir de los clasificadores definidos en la Taxonomía de la Actividad (TxA). Se pretende encontrar una correspondencia entre los clasificadores definidos en la Taxonomía y los elementos de modelado seleccionados. La evaluación está orientada también a identificar los elementos de modelado más utilizados por la comunidad de expertos en Ingeniería de Software (IS) y en Interacción Humano Computador (HCI).

6.4.2 Proyectos de investigación

- Participación como asesora en el proyecto: **"Framework para la ejecución de modelos ejecutables específicos del dominio"**: Esta propuesta está orientada a fortalecer las capacidades en investigación en Model-driven Engineering del grupo de investigación SINFOCI de la Universidad del Quindío, mediante procesos de investigación científica en el área de calidad de lenguajes de modelado y notaciones, la extensión de marcos de desarrollo y la mejora en la expresividad de lenguajes, con el propósito de consolidar las líneas de investigación del grupo usando un enfoque dirigido por modelos que permita establecer lineamientos conceptuales, metodológicos y tecnológicos para la generación de procesos y productos en los diferentes proyectos de investigación e innovación desarrollados por el grupo. El producto principal del proyecto es una herramienta software que optimiza la generación de código en el desarrollo de los sistemas interactivos.

Proyecto enmarcado dentro del programa de Formación de Jóvenes Investigadores, patrocinado por Colciencias y la Universidad del Quindío, año 2014.

6.4.3 Pasantías y estancias de investigación

- Pasantía Internacional Benemérita Universidad Autónoma de Puebla en México (BUAP), febrero – Agosto 2015.
- Participación como profesor invitado, en el marco de la pasantía en México. Universidad Popular Autónoma del Estado de Puebla (UPAEP). Conferencia: *"Taxonomía de la Actividad: Análisis de Propuestas de Desarrollo de Sistemas Interactivos"*.
- Beca de movilidad: AUJP Estancia investigativa en la Universidad de Granada, España. Septiembre 8 al 18. 2016.





REFERENCIAS





REFERENCIAS

A

(Abran et al, 2004)

Abran, A., Moore, J. W., Bourque, P., Dupuis, R., Tripp, L. (2004). SWEBOK: Guide to the Software Engineering Body of Knowledge: IEEE computer Society Press.

(Ambler et al, 2005)

Ambler, S. W., Nalbene, J., Vizdos, M. (2005). Extending The RUP With The Zachman Framework: Pearson Education.

(Anda y Sjøberg, 2002)

Anda, B., Sjøberg, D. I. K. (2002). Towards an inspection technique for use case models. Paper presented at the Proceedings of the 14th international conference on Software engineering and knowledge engineering, Ischia, Italy.

(Andova et al, 2012)

Andova, S., van den Brand, M. G. J., Engelen, L. J. P., Verhoeff, T. (2012). MDE Basics with a DSL Focus. In M. Bernardo, V. Cortellessa, A. Pierantonio (Eds.), Formal Methods for Model-Driven Engineering: 12th International School on Formal Methods for the Design of Computer, Communication, and Software Systems, SFM 2012, Bertinoro, Italy, June 18-23, 2012. Advanced Lectures (pp. 21-57). Berlin, Heidelberg: Springer Berlin Heidelberg.

(Annett y Duncan, 1967)

Annett, J., Duncan, K. D. (1967). Task Analysis and Training Design. *Occupational Psychology*, 41, 211-221.

(Arteaga et al, 2015)

Arteaga, J. M., Calleros, J. M. G., Huitrón, A. S. (2015). La Interacción Humano-Computadora en México. Naucalpan de Juárez, Estado de México: Pearson Educación.

(Atkinson y Kühne, 2001)

Atkinson, C., Kühne, T. (2001). The Essence of Multilevel Metamodeling Paper presented at the Proceedings of the 4th International Conference on The Unified Modeling Language, Modeling Languages, Concepts, and Tools.

B

(Balbo et al, 2004)

Balbo, S., Ozkan, N., Paris, C. (2004). Choosing the right task modelling notation: A taxonomy. In D. Diaper N. A. Stanton (Eds.), *The handbook of task analysis for human-computer interaction*. Mahwah, New Jersey: Lawrence Erlbaum Associates.

(Balduino, 2007)

Balduino, R. (2007). Introduction to OpenUP (Open Unified Process). Eclipse site. www.eclipse.org/epf/general/OpenUP.pdf

(Barboni et al, 2010)

Barboni, E., Ladry, J.-F., Navarre, D., Palanque, P., Winckler, M. (2010, June 19–23, 2010). Beyond Modelling: An Integrated Environment Supporting Co-Execution of Tasks and Systems Models. Paper presented at the EICS'10, Berlin, Germany.

(Bernárdez et al, 2004)

Bernárdez, B., Durán, A., Genero, M. (2004). Empirical Evaluation and Review of a Metrics-Based Approach for Use Case Verification. *Journal of Research and Practice in Information Technology*, 36(4).

(Berti et al, 2004)

Berti, S., Mori, G., Paternò, F., Santoro, C. (2004). A transformation-based environment for designing multi-device interactive applications. Paper presented at the Proceedings of the 9th international conference on Intelligent user interfaces, Funchal, Madeira, Portugal.

(Bodart y Vanderdonckt, 1995)

Bodart, F., Vanderdonckt, J. (1995). Towards a Systematic Building of Software Architectures: the TRIDENT Methodological Guide. Paper presented at the Design, Specification and Verification of Interactive Systems, DSV-IS'95.

(Bolognesi y Brinskma, 1987)

Bolognesi, T., Brinskma, E. (1987). Introduction to the ISO Specification Language LOTOS. *Computer Networks and ISDN Systems*, 14(1), 25-59.

(Booch et al, 1998)

Booch, G., Rumbaugh, J., Jacobson, I. (1998). *The Unified Modeling Language User Guide* (1 ed.): Addison Wesley Longman, Inc.

(Budinsky et al, 2003)

Budinsky, F., Steinberg, D., Merks, E., Ellersick, R., Grose, T. J. (2003). *Eclipse Modeling Framework: A Developer's Guide*: Addison Wesley.

C

(Cesare y Serrano, 2006)

Cesare, S. d., Serrano, A. (2006). Collaborative Modeling Using UML and Business Process Simulation Paper presented at the Proceedings of the 39th Annual Hawaii International Conference on System Sciences - Volume 01.



(Cheng-Leong et al, 1999)

Cheng-Leong, A., Pheng, K., Leng, G. (1999). IDEF*: a comprehensive modelling methodology for the development of manufacturing enterprise systems. *International Journal of Production Research*, 37(17), 3839-3858.

(Cleophas et al, 2005)

Cleophas, L., Watson, B. W., Kourie, D. G., Boake, A. (2005). TABASCO: a taxonomy-based domain engineering method Paper presented at the Proceedings of the 2005 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries White River, South Africa

(Cockburn, 2000)

Cockburn, A. (2000). *Writing Effective Use Cases*: Addison-Wesley.

(Collins et al, 2002)

Collins, P., Shukla, S., Redmiles, D. (2002). Activity Theory and System Design: A View from the Trenches. *Computer Supported Cooperative Work (CSCW)*, 11(1-2), 55-80. doi: 10.1023/A:1015219918601

(Constantine, 2006)

Constantine, L. L. (2006). Activity Modeling: Toward a Pragmatic Integration of Activity Theory with Usage-Centered Design.

(Constantine, 2009)

Constantine, L. (2009). Human Activity Modeling: Toward A Pragmatic Integration of Activity Theory and Usage-Centered Design. In A. Seffah, J. Vanderdonckt, M. Desmarais (Eds.), *Human-Centered Software Engineering* (pp. 27-51): Springer London

(Constantine y Lockwood, 1999)

Constantine, L. L., Lockwood, L. A. D. (1999). *Software For Use: A Practical Guide to the Models and Methods of Usage-Centered Design*: Addison-Wesley.

(Constantine y Lockwood, 2001)

Constantine, L. L., Lockwood, L. A. D. (2001). Structure and style in use cases for user interface design *Object modeling and user interface design: designing interactive systems* (pp. 245-279): Addison-Wesley Longman Publishing Co., Inc.

(Constantine et al, 2003)

Constantine, L., Windl, H., Noble, J., Lockwood, L. (2003). *From Abstraction to Realization: Canonical Abstract Prototypes for User Interface Design REVISED*.



(Corporation, 2015)

Corporation, Oracle. Java, Recursos de Ayuda. 2015, from <https://www.java.com/es/about/>

D

(Diaper, 1989)

Diaper, D. (1989). The discipline of human-computer interaction. *Interacting with Computers*, 1(1), 117-138.

E

(Engeström, 1987)

Engeström, Y. (1987). Learning by Expanding: An activity-theoretical approach to developmental research: Orienta-Konsultit Oy.

(Engeström, 2001)

Engeström, Y. (2001). Expansive Learning at Work: toward an activity theoretical reconceptualization. *Journal of Education and Work*, 14(1). doi: 10.1080/13639080020028747

(España et al, 2009)

España, S., González, A., Pastor, Ó. (2009). Communication Analysis: A Requirements Engineering Method for Information Systems. In P. van Eck, J. Gordijn, R. Wieringa (Eds.), *Advanced Information Systems Engineering: 21st International Conference, CAiSE 2009, Amsterdam, The Netherlands, June 8-12, 2009. Proceedings* (pp. 530-545). Berlin, Heidelberg: Springer Berlin Heidelberg

(España, 2011)

España, S. (2011). Methodological Integration of Communication Analysis into a Model-Driven Software Development Framework. (Doctorado), Universidad Politécnica de Valencia.

F

(Favre, 2005)

Favre, J. M. (2005). Foundations of Meta-Pyramids: Languages vs. Metamodels, Episode II: Story of Thotus the Baboon1. Paper presented at the Dagstuhl Seminar on Model Driven Approaches for Language Engineering.

(Ferré, 2001)

Ferré, X., Juristo, N., Windl, H., Constantine, L. (2001). Usability Basics for Software Developers. *IEEE Softw.*, 18(1), 22-29. doi: 10.1109/52.903160



(France y Rumpe 2007)

France, R., Rumpe, B. (2007). Model-driven Development of Complex Software: A Research Roadmap. Paper presented at the 2007 Future of Software Engineering.

(Frankel, 2004a)

Frankel, D. S. (2004). An MDA Manifesto. MDA Journal. www.bptrends.com/publicationfiles/05-04%20COL%20IBM%20Manifesto%20-%20Frankel%20-3.pdf

(Frankel, 2004b)

Frankel, D. S. (2004). Domain-specific modeling and model driven architecture. MDA Journal.

(Frankel et al, 2003)

Frankel, D. S., Harmon, P., Mukerji, J., Odell, J., Owen, M., Rivitt, P., . . . Soley, R. M. (2003). The Zachman Framework and the OMG's Model Driven Architecture. MDA Journal. <http://www.bptrends.com/publicationfiles/09%2D03%20WP%20Mapping%20MDA%20to%20Zachman%20Framework%2Epdf>

G

(Garrido, 2003)

Garrido, J. L. (2003). AMENITIES: Una metodología para el desarrollo de sistemas cooperativos basada en modelos de comportamiento y tareas. (PhD.), Universidad de Granada, Granada.

(Gea et al, 2003)

Gea, M., Gutierrez, F. L., Garrido, J.L., C, J. (2003). Teorías y modelos conceptuales para un diseño basado en grupos. Paper presented at the IV Congreso Internacional de Interacción Persona-Ordenador, Interacción'2003, Vigo.

(Georg, 2011)

Georg, G. (2011). Activity Theory and its Applications in Software Engineering and Technology (pp. 22). Fort Collins, Colorado: Colorado State University.

(Georg et al, 2015)

Georg, G., Mussbacher, G., Amyot, D., Petriu, D., Troup, L., Lozano-Fuentes, S., France, R. (2015). Synergy between Activity Theory and goal/scenario modeling for requirements elicitation, analysis, and evolution. *Information and Software Technology*, 59(0), 109-135. doi: <http://dx.doi.org/10.1016/j.infsof.2014.11.003>

(Georg y Troup, 2013)

Georg, G., Troup, L. (2013). Experiences Developing a Requirements Language Based on the Psychological Framework Activity Theory. Paper presented at the OCL@MoDELS.

(Giese et al 2008)

Giese, M., Mistrzyk, T., Pfau, A., Szwillus, G., von Detten, M. (2008). AMBOSS: A Task Modeling Approach for Safety-Critical Systems. In P. Forbrig F. Paternò (Eds.), Engineering Interactive Systems: Second Conference on Human-Centered Software Engineering, HCSE 2008, and 7th International Workshop on Task Models and Diagrams, TAMODIA 2008, Pisa, Italy, September 25-26, 2008. Proceedings (pp. 98-109). Berlin, Heidelberg: Springer Berlin Heidelberg

(Giraldo, 2000)

Giraldo, W. J. (2000). Diseño de un Ejecutivo Multitarea en Tiempo Real. (Maestría), Universidad del Valle, Cali, Valle.

(Giraldo, 2010)

Giraldo, W. J. (2010). Marco de Desarrollo de Sistemas Software Interactivos Basado en la Integración de Procesos y Notaciones. (Phd Doctoral), Universidad de Castilla-La Mancha, Ciudad Real.

(Giraldo et al, 2001)

Giraldo, W. J., Villegas, M. I., Candela, C., Castro, L. M. (2001). Software de Automatización Industrial (SAI), Concepción y Desarrollo. Revista de Investigaciones, 3, 112-122.

(Giraldo et al, 2004)

Giraldo, W. J., Villegas, M. L., Marín, D. F. (2004). Diseño y construcción de un software para automatización de procesos SAI (Vol. 1). Cali: Impactar S.A.

(Giraldo et al, 2014a)

Giraldo, W. J., Tobón, M. L., Giraldo, F. D., Villegas, M. L., Guerrero, A., Cortés, M. Y., . . . Collazos, C. A. (2014a). HCI Incorporation: a case for Colombia. Paper presented at the Proceedings of the XV International Conference on Human Computer Interaction, Puerto de la Cruz, Tenerife, Spain.

(Giraldo et al, 2014b)

Giraldo, W. J., Villegas, M. L., Collazos, C. A. (2014b), (3-5 Sept. 2014). Incorporation of HCI: Classification of activity modeling. Paper presented at the Computing Colombian Conference (9CCC), 2014 9th.



(González et al, 2011)

González, A., Ruiz, M., España, S., Oscar Pastor. (2011). Message Structures: a modelling technique for information systems analysis and design. Paper presented at the 14th Workshop on Requirements Engineering (WER 2011), Rio de Janeiro, Brazil.

(Göransson et al, 2003)

Göransson, B., Lif, M., Gulliksen, J. (2003). Usability Design—Extending Rational Unified Process With A New Discipline. Paper presented at the Lecture Notes in Computer Science.

(Gottesdiener, 2002)

Gottesdiener, E. (2002). Top Ten Ways Project Teams Misuse Use Cases -- and How to Correct Them. Rational Edge.

(Guerrero, 2010)

Guerrero, J. (2010). A Methodology for Developing User Interfaces to Workflow Information Systems. (Doctor of Philosophy in Economics and Management Sciences), Université catholique de Louvain.

(Graham, 1996)

Graham, I. (1996). Task Scripts, Use Cases and Scenarios in Object Oriented Analysis. Object-Oriented Systems, 123--142.

(Granollers, 2004)

Granollers, T. (2004). Mpiu+A. Una Metodología que Integra la Ingeniería del Software, La Interacción Persona Ordenador y la Accesibilidad en el Contexto de Equipos de Desarrollo Multidisciplinares. (Doctorate), Lleida.

(Gronback, 2009)

Gronback, R. C. (2009). ECLIPSE MODELING PROJECT, A Domain-Specific Language Toolkit. Boston: Addison Wesley.

(Group, 2005)

Group, A., INRIA, L., Nantes. (2005). Atlas Transformation Language: ATL Starter's Guide. Retrieved 10-08, 2007.

H

(Haan et al, 1991)

Haan, G. d., Veer, G. C. v. d., Vliet, J. C. v. (1991). Formal modelling techniques in human-computer interaction. Acta Psychologica, 78(1-3), 27-67.

(Harel, 1987)

Harel, D. (1987). Statecharts: a visual formalism for complex systems. Science of Computer Programming, 8(3), 231-274.

(Hasan, 1998)

Hasan, H. (1998). Integrating IS and HCI using activity theory as a philosophical and theoretical basis. *Australian Journal of Information Systems*, 6(2), 44-55.

(Haynes et al, 2004)

Haynes, S. R., Puroo, S., Skattebo, A. L. (2004). Situating evaluation in scenarios of use. Paper presented at the Proceedings of the 2004 ACM conference on Computer supported cooperative work, Chicago, Illinois.

(Hornbæk et al, 2007)

Hornbæk, K., Høegh, R., Pedersen, M., Stage, J. (2007). Use Case Evaluation (UCE): A Method for Early Usability Evaluation in Software Development. In C. Baranauskas, P. Palanque, J. Abascal, S. Barbosa (Eds.), *Human-Computer Interaction – INTERACT 2007* (Vol. 4662, pp. 578-591): Springer Berlin Heidelberg.

I

(IBM_Rational, 2003a)

IBM_Rational. (2003a). Too Navigator (Rational Unified Process).

(IBM_Rational 2003b)

IBM_Rational. (2003b). Too Navigator (Rational Unified Process), Concepts: User-Centered Design.

(ICONIX, 2013)

ICONIX Software Engineering, I. (2013). ICONIX JumpStart Training. Retrieved 5 mayo de 2016, from <http://www.iconixsw.com/jumpstart.html>

(Iqbal et al, 2005)

Iqbal, R., James, A., Gatward, R. (2005). Designing with ethnography: An integrative approach to CSCW design. *Advanced Engineering Informatics*, 19(2), 81-92.

J

(Jacobson, 1992)

Jacobson, I. (1992). *Object Oriented Software Engineering: A Use Case Driven Approach*: Addison Wesley.

(Jacobson et al, 2011)

Jacobson, I., Spence, I., Bittner, K. (2011). Use-Case 2.0. The Guide to Succeeding with Use Cases. <http://www.pmi.org/~media/PDF/Knowledge%20Center/IJIUseCase20.ashx>.



(Jagielska et al, 2006)

Jagielska, D., Wernick, P., Wood, M., Bennett, S. (2006). How natural is natural language?: how well do computer science students write use cases? Paper presented at the Companion to the 21st ACM SIGPLAN symposium on Object-oriented programming systems, languages, and applications, Portland, Oregon, USA.

(James, 2013)

James, M. (2013). Scrum Methodology. Retrieved 5 de mayo de 2016, from <http://scrummethodology.com/>

(Jouault et al, 2006)

Jouault, F., Allilaire, F., Bézivin, J., Kurtev, I., Valduriez, P. (2006). ATL: a QVT-like transformation language. Paper presented at the Companion to the 21st ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications Portland, Oregon.

(Jouault y Kurtev, 2006)

Jouault, F., Kurtev, I. (2006). On the architectural alignment of ATL and QVT Paper presented at the Proceedings of the 2006 ACM symposium on Applied computing, Dijon, France.

K

(Kaptelinin et al, 1999)

Kaptelinin, V., Nardi, B. A., Macaulay, C. (1999). Methods & tools: The activity checklist: a tool for representing the "space" of context. *Interactions*, 6(4), 27-39. doi: 10.1145/306412.306431

(Kim et al, 2007)

Kim, T.-Y., Lee, S., Jeong-Soo Lee, K. K., and Cheol-Han Kim. (2007). Enterprise Architecture Framework for Agile and Interoperable Virtual Enterprises. In P. Saha (Ed.), *Enterprise systems architecture in practice* (Vol. 1, pp. 62-84): Idea Group Inc.

(Kitchenham, 2004)

Kitchenham, B. (2004). Procedures for Performing Systematic Reviews, TR/SE-0401.

(Korpela et al, 2000)

Korpela, M., Soriyan, H. A., Olufokunbi, K. C. (2000). ACTIVITY ANALYSIS AS A METHOD FOR INFORMATION SYSTEMS DEVELOPMENT: General Introduction and Experiments from Nigeria and Finland. *Scandinavian Journal of Information Systems*, 12(1).

(Kramer, 2007)

Kramer, J. (2007). Is Abstraction The Key to Computing? *Communications of ACM*, 50(4).



(Kramer y Hazzan, 2006)

Kramer, J., Hazzan, O. (2006). The Role of Abstraction in Software Engineering. Paper presented at the 28th International Conference on Software Engineering (ICSE) Shanghai, China.

(Küne, 2005)

Küne, T. (2005). Understanding metamodeling. Paper presented at the Proceedings of the 27th international conference on Software engineering St. Louis, MO, USA

(Kuutti, 1991)

Kuutti, K. (1991). The concept of activity as a basic unit of analysis for CSCW research. Paper presented at the Second European Conference on Computer Supported Cooperative Work (ECSCW'91).

(Kwasnik, 1992)

Kwasnik, B. H. (1992). The Role of Classification Structures in Reflecting and Building Theory.

L

(Langlands y Edwards, 2009)

Langlands, M., Edwards, C. (2009, 22/02/2009). Business vs. System Use Cases. AgileEA.com.

(Lauesen, 2005)

Lauesen, S. (2005). User Interface Design: A Software Engineering Perspective. Harlow: Addison-Wesley.

(Lessiter et al, 2014)

Lessiter, J., Freeman, J., Miotto, A., Ferrari, E. (2014). Ghosts in the Machines: Towards a Taxonomy of Human Computer Interaction. Springer International Publishing Switzerland 2014, 21–31.

(Letelier et al, 1998)

Letelier, P., Sánchez, P., Ramos, I., Pastor, O. (1998). OASIS 3.0: Un enfoque formal para el modelado conceptual orientado a objeto: Universidad Politécnica de Valencia SPUPV-98.4011

(Limbourg, 2004)

Limbourg, Q. (2004). Multi-Path Development of User Interfaces. (Phd.), Université catholique de Louvain, Louvain-la-Neuve.

(Limbourg y Vanderdonckt, 2004a)

Limbourg, Q., Vanderdonckt, J. (2004a). Comparing Task Models for User Interface Design. In D. Diaper N. Stanton (Eds.), The handbook of task analysis for HCI. Mahwah, New Jersey: Lawrence Erlbaum Associates



(Limbourg y Vanderdonckt, 2004b)

Limbourg, Q., Vanderdonckt, J. (2004b). UsiXML: A User Interface Description Language Supporting Multiple Levels of Independence. In M. M. C. S (Eds.), *Engineering Advanced Web Applications* (pp. 325-338): Rinton Press, Paramus

(Liskov y Zilles, 1975)

Liskov, B., Zilles, S. (1975). Specification techniques for data abstractions. Paper presented at the Proceedings of the international conference on Reliable software.

(Lozano, 2001)

Lozano, M. D. (2001). *Entorno Metodológico Orientado a Objetos para la Especificación y Desarrollo de Interfaces de Usuario*. (Phd.), Universidad Politecnica de Valencia.

(Luyten, 2004)

Luyten, k. (2004). *Dynamic User Interface Generation for Mobile and Embedded Systems with Model-Based User Interface Development*. (PhD. thesis,), Universiteit Limburg.

M

(Meixner y Seissler, 2011)

Meixner, G., Seissler, M. (2011). Selecting the Right Task Model for Model-based User Interface Development. Paper presented at the The Fourth International Conference on Advances in Computer-Human Interactions (ACHI 2011).

(MetaCase, 2005)

MetaCase. (2005). Domain-specific modelling. 08-07-2007, from <http://www.appdevadvisor.co.uk/express/vendor/domain.html>

(Miller y Mukerji, 2003)

Miller, J., Mukerji, J. (2003). MDA Guide Version 1.0.1. 08-07-2007, from <http://www.omg.org/docs/omg/03-06-01.pdf>

(Microsoft, 2017)

Microsoft, N. (2017). The MVVM Pattern. Retrieved 14-feb-2017, 2017, from <https://msdn.microsoft.com/en-us/library/hh848246.aspx>

(Molina, 2006)

Molina, A. I. (2006). *Una Propuesta Metodológica Para El Desarrollo De La Interfaz De Usuario En Sistemas Groupware (CIAM)*. (Doctorate), Castilla-La Mancha, Ciudad Real.

(Molina et al, 2006)

Molina, A. I., Redondo, M. A., Ortega, M. (2006). CIAM: Una Aproximación Metodológica para el desarrollo de Interfaces de Usuario en aplicaciones groupware. Paper presented at the VII Congreso Internacional de Interacción Persona-Ordenador. Puertollano (Spain).

(Molina et al, 2008)

Molina, A. I., Redondo, M. A., Ortega, M., Hope, U. (2008). CIAM: A methodology for the development of groupware user interfaces. Journal of Universal Computer Science(JUCS), 14(9).

(Molina et al, 2009)

Molina, A. I., Redondo, M. A., Ortega, M. (2009). A Review of Notations for Conceptual Modeling of Groupware Systems. In J. A. Macías, T. Granollers, P. M. Latorre (Eds.), New Trends on Human-Computer Interaction (pp. 75-86): Springer London

(Moody, 2010)

Moody, D. L. (2010). The "physics" of notations: a scientific approach to designing visual notations in software engineering. Paper presented at the Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - Volume 2, Cape Town, South Africa.

(Moore et al, 2004)

Moore, B., Dean, D., Gerber, A., Wagenknecht, G., Vanderheyden, P. (2004). Eclipse Development using the Graphical Editing Framework and the Eclipse Modeling Framework: ibm.com/redbooks.

(Moran, 1981)

Moran, T. P. (1981). The command language grammar: a representation for the user interface of interactive systems. International Journal of man-machine studies, 15(1), 3-50.


N

(Nardi, 1996)

Nardi, B. A. (1996). Activity Theory and Human-Computer Interaction. In MIT Press (Ed.), Context and Consciousness: Activity Theory and Human-Computer Interaction (pp. 5). Cambridge, MA.

(Navarre et al, 2009)

Navarre, D., Palanque, P., Winckler, M. (2009). Task Models and System Models as A Bridge Between Hci and Software Engineering. In A. Seffah, J. Vanderdonckt, M. C. Desmarais (Eds.), Human-Centered Software Engineering: Software Engineering Models, Patterns and Architectures for HCI (pp. 357-385). London: Springer London.

- 
- (Neto et al, 2005a)
Neto, G. C., Gomes, A. S., Castro, J. B. d. (2005a). Mapping Activity Theory Diagrams into i* Organizational Models. JCS&T, 5(2).
- (Neto et al, 2005b)
Neto, G. C., Gomes, A. S., Castro, J., Sampaio, S. (2005b). Integrating activity theory and organizational modeling for context of use analysis. Paper presented at the Proceedings of the 2005 Latin American conference on Human-computer interaction, Cuernavaca, Mexico.
- (Ng, 2002)
Ng, P.-W. (2002). Effective Business Modeling with UML: Describing Business Use Case and Realizations. The Rational Edge.
- (Nielsen, 1993)
Nielsen, J. (1993). Usability Engineering. San Francisco: Morgan Kaufmann.
- (Nielsen, 1994)
Nielsen, J. (1994). Heuristic evaluation. In J. Nielsen R. L. Mack (Eds.), Usability Inspection Methods. New York: John Wiley & Sons
- (Nunes, 2001)
Nunes, D. N. J. (2001). Object Modeling for User-Centered Development and User Interface Design: The Wisdom Approach. (phd Doctorate), Universidade da Madeira, Funchal.
- O**
- (Olivier et al, 2004)
Olivier, D., Bertrand, D., Rene, C. (2004). Task modelling for capillary collaborative systems based on scenarios. Paper presented at the Proceedings of the 3rd annual conference on Task models and diagrams, Prague, Czech Republic.
- (OMG, 1999)
OMG. (1999). Unified Modeling Language 1.3, Standard. Retrieved 12-10, 2007, from <http://www.omg.org/spec/UML/1.3/>
- (OMG, 2008a)
OMG. (2008a). Software & Systems Process Engineering Metamodel specification (SPEM) Version 2.0. Retrieved 11-2008, from <http://www.omg.org/spec/SPEM/2.0/>
- (OMG, 2008b)
OMG. (2008b). Unified Modeling Language (UML) Version 2.0. Retrieved 11-2008, from <http://www.uml.org/>

(OMG, 2011)

OMG. (2011). Business Process Model and Notation (BPMN) (Version 2.0 ed.): Object Management Group (OMG).

(OMG, 2012)

OMG. (2012). Business Process Management Initiative. Retrieved 22-jul-2016, from <http://web.archive.org/web/20120103112152/http://bpmn.org/Documents/FAQ.htm>

(ORMS, 2001)

ORMS, Object and Reference Model Subcommittee of the Architecture Board. (2001). A Proposal for an MDA Foundation Model: OMG. WhitePaper

(Oxton, 2015)

Oxton, G., Chmaj, J., Kay, D. (2015). Perspectives on Taxonomy, Classification, Structure and Find-ability.

P

(Pastor et al, 2001)

Pastor, O., Gómez, J., Insfrán, E., Pelechano, V. (2001). The OO-method approach for information systems modeling: from object-oriented conceptual modeling to automated programming. *Inf. Syst.*, 26(7), 507-534. doi: 10.1016/S0306-4379(01)00035-7

(Paternò, 2001)

Paternò, F. (2001). Towards a UML for Interactive Systems. Paper presented at the 8th International Conference on Engineering for Human-Computer Interaction.

(Paternò, 2004)

Paternò, F. (2004). ConcurTaskTrees: An Engineered Notation for Task Models. *The Handbook Of Task Analysis For HCI*.

(Paternò et al, 1997)

Paternò, F., Mancini, C., Meniconi. (1997). ConcurTaskTree: A diagrammatic notation for specifying task models. Paper presented at the IFIP TC 13 International Conference on Human-Computer Interaction Interact'97, Sydney.

(Paternò et al, 1998)

Paternò, F., Santoro, C., Tahmassebi, S. (1998). Formal model for cooperative tasks: Concepts and an application for en-route air traffic control. Paper presented at the 5th Int. Workshop on Design, Specification, and Verification of Intractive Systems DSV-IS '98, Abingdon.



(Penichet, 2007)

Penichet, V. M. R. (2007). Modelo de Proceso para el Desarrollo de Interfaces en Entornos CSCW Centrado en los Usuarios y Dirigido por Tareas. (Phd), Castilla-La Mancha, Albacete.

R

(RAE, 2015)

RAE. Real Academia Española. (Ed.) (2015).

(Rational, 1997)

Rational Software. (1997). Unified Modeling Language (UML). UML Semantics, Version 1.1.

(Rech y Bunse, 2009)

Rech, J., Bunse, C. (2009). Model-Driven Software Development: Integrating Quality Assurance. Hershey, New York: Information Science Reference.

(Rosson y Carroll, 2002)

Rosson, M. B., Carroll, J. M. (2002). Usability Engineering: Scenario-Based Development of Human Computer Interaction: Morgan Kaufmann.

(Rourke, 2002)

Rourke, C. (2002). Making UML the lingua Franca of usable system design. Interfaces Magazine from the British HCI Group.

S

(Scott y Stanley, 2001)

Scott, E. D., Stanley, G. S. (2001). Successful Software Development (2nd ed.): Prentice-Hall PTR.

(Schwaber y Sutherland, 2010)

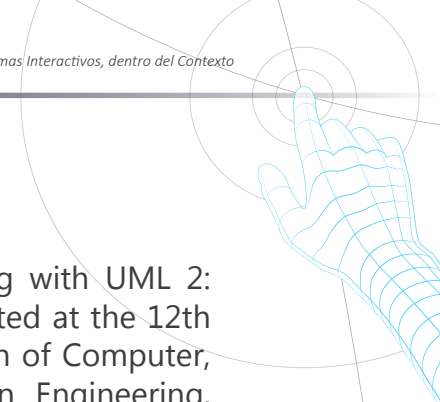
Schwaber, K., Sutherland, J. (2010). Scrum: Developed and sustained: Scrum.org.

(Seffah et al, 2001)

Seffah, A., Djouab, R., Antunes, H. (2001). Comparing and reconciling usability-centered and use case-driven requirements engineering processes. Aust. Comput. Sci. Commun., 23(5), 132-139.

(Sekaran, 2006)

Sekaran, U. (2006). Research Methods for Business: A Skill Building Approach. New York: Wiley India.



(Selic, 2012)

Selic, B. (2012). Elements of Model-Based Engineering with UML 2: What They Don't Teach You About UML. Paper presented at the 12th International School on Formal Methods for the Design of Computer, Communication and Software Systems: Model-Driven Engineering, Bertinoro, Italy.

(Shneiderman, 2004)

Shneiderman, B. (2004). Designing the User Interface: Strategies for Effective Human-Computer Interaction. (4 ed.). USA: Addison-Wesley, Reading (MA).

(SIGCHI, 1992)

SIGCHI, A. (1992). Curricula for Human-Computer Interaction. ACM Press, 1992.

(Sjoberg et al, 2007)

Sjoberg, D. I. K., Dyba, T., Jorgensen, M. (2007). The Future of Empirical Methods in Software Engineering Research. Paper presented at the 2007 Future of Software Engineering.

(Souza, 2003)

Souza, C. R. B. d. (2003). Interpreting Activity Theory as a Software Engineering Methodology. Paper presented at the Second European Conference on Computer Supported Cooperative Work (ECSCW'03).

(Souza, 2005)

Souza, C. S. d. (2005). The Semiotic Engineering of Human-Computer Interaction (Acting with Technology): The MIT Press.

(Sowa y Zachman, 1992)

Sowa, J. F., Zachman, J. A. (1992). Extending and formalizing the framework for information systems architecture IBM Syst. J, 590-616

(Spangler et al, 2003)

Spangler, S., Kreulen, J. T., Lessler, J. (2003). Generating and Browsing Multiple Taxonomies Over a Document Collection. J. Manage. Inf. Syst, 19(4), 191-212.

(Stuart et al, 1983)

Stuart, K. C., Allen, N., Thomas, P. M. (1983). The Psychology of Human-Computer Interaction: L. Erlbaum Associates Inc.

(Sutcliffe, 2014)

Sutcliffe, A. G. (2014). Requirements Engineering. The Encyclopedia of Human-Computer Interaction, 2nd Ed.

T

(Tamayo, 1999)

Tamayo, M. T. (1999). El Proyecto De Investigación (Vol. 5). Bogotá.

(Trætteberg, 2002)

Trætteberg, H. (2002). Model-based User Interface Design. (Phd.), Norwegian University of Science and Technology.

(Tzitzikas et al, 2005)

Tzitzikas, Y., Spyratos, N., Constantopoulos, P. (2005). Mediators over taxonomy-based information sources. The VLDB Journal, 14(1), 112-136.

U

(Unterkalmsteiner et al, 2014)

Unterkalmsteiner, M., Feldt, R., Gorschek, T. (2014). A taxonomy for requirements engineering and software test alignment. ACM Transactions on Software Engineering and Methodology (TOSEM), 23(2), 16.

V

(Villegas et al, 2009)

Villegas, M. L., Castro, L. M., Ruiz, A. (2009). UML Semantics v 1.1 - Implementación del Metamodelo Sobre EMTRSAI. Armenia, Quindío. Colombia: Arte Imagen.

(Villegas et al, 2015)

Villegas, M. L., Collazos, C. A., Giraldo, W. J., Gonzalez, J. M. (2015). Incorporación de HCI: Validación de la Usabilidad en Casos de Uso mediante la Taxonomía de la Actividad. Paper presented at the 10 Congreso Colombiano de Computación (10CCC), Bogotá, Colombia.

(Villegas et al, 2016a)

Villegas, M. L., Collazos, C. A., Giraldo, W. J., Gonzalez, J. M. (2016a). Activity Theory as a Framework for Activity Taxonomy in HCI. IEEE Latin America Transactions, 14(2), 844 - 857. doi: 10.1109/TLA.2016.7437231

(Villegas et al, 2016b)

Villegas, M. L., Collazos, C. A., Giraldo, W. J., Gonzalez, J. M., Guerrero, J. (2016b). Taxonomía de la Actividad: Selección de Elementos de Modelado para la Especificación de Sistemas Interactivos. Paper presented at the Interacción 2016. XVII International Conference on Human Computer Interaction, Salamanca, España.

(Vygotski, 1980)

Vygotski, L. (1980). *Mind in Society: Development of Higher Psychological Processes* (H. U. Pr Ed.).

W

(Wells, 2013)

Wells, D. (2013). *Extreme Programming: A gentle introduction*. Retrieved 5 mayo de 2016, 2016, from <http://www.extremeprogramming.org/>

(WFMC, 1999)

WFMC. (1999). *Workflow Management Coalition. Terminology & Glossary (WFMC-TC-1011)* (pp. 65). Hampshire, United Kingdom: Workflow Management Coalition.

(Wing, 2006)

Wing, J. M. (2006). *Computational Thinking*. *Communications of ACM*, 49(3).

Y

(Yin, 2003)

Yin, R. K. (2003). *Applications of Case Study Research* (2 ed.). London: Sage Publications Inc.

Z

(Zachman, 1987)

Zachman, J. A. (1987). *A Framework For Information Systems Architecture*. *IBM Ssystems Journal*, 26(3).

(Zachman, 2007-2011)

Zachman, J. A. (2007-2011). *The Zachman Framework for Enterprise Architecture - The Enterprise Ontology* (3.0 ed.). www.zachman.com.

(Zhang y Lee, 2004)

Zhang, D., Lee, W. S. (2004). *Web taxonomy integration using support vector machines*. Paper presented at the Proceedings of the 13th international conference on World Wide Web, New York, NY, USA



ANEXOS

ANEXOS

ANEXO A: EVALUACIÓN DE NOTACIONES PARA EL MODELADO DE TAREAS - PERFIL DE LOS ENCUESTADOS

ANEXO B: EVALUACIÓN DE MODELADO DE TAREAS PARA SISTEMAS INTERACTIVOS - CONTEXTO

ANEXO C: SELECCIÓN DE ELEMENTOS DE MODELADO DE SISTEMAS INTERACTIVOS

ANEXO D: NOTACIÓN QUE SOPORTA EL MODELADO DE LA ACTIVIDAD PARA AMBUID

ANEXO E: METAMODELO DEFINIDO PARA AMBUID

ANEXO F: MODELO DE DISEÑO DE EJECUCIÓN DE MÁQUINAS DE ESTADO

ANEXO A

EVALUACIÓN DE NOTACIONES PARA EL MODELADO DE TAREAS PERFIL DE LOS ENCUESTADOS

Con esta encuesta se pretende clasificar los perfiles de las personas encuestadas en el proceso de someter a evaluación un conjunto de notaciones utilizadas en el desarrollo de sistemas interactivos, específicamente en el modelado de la actividad.

1. Nombre del encuestado: _____

2. ¿En qué ámbito laboral se desempeña actualmente? (*marque sólo una opción*):

Empresarial <input type="radio"/>	Académico <input type="radio"/>	Empresarial y académico <input type="radio"/>	Otro <input type="radio"/>
-----------------------------------	---------------------------------	---	----------------------------

3. ¿Cuál es su mayor nivel de titulación? (*marque sólo una opción*):

Doctorado <input type="radio"/>	Maestría <input type="radio"/>	Especialización <input type="radio"/>	Pregrado <input type="radio"/>
---------------------------------	--------------------------------	---------------------------------------	--------------------------------

4. ¿En qué área obtuvo su mayor titulación?: _____

5. ¿Durante su formación, cuántos cursos del área de Ing. de Software ha visto? (*marque sólo una opción*):

Entre 1 y 3 <input type="radio"/>	Más de 3 <input type="radio"/>	Ninguno <input type="radio"/>
-----------------------------------	--------------------------------	-------------------------------

6. ¿Durante su formación, cuántos cursos del área de Interacción Humano Computador (HCI) ha visto? (*marque sólo una opción*):

Entre 1 y 3 <input type="radio"/>	Más de 3 <input type="radio"/>	Ninguno <input type="radio"/>
-----------------------------------	--------------------------------	-------------------------------

7. ¿Se considera usted experto en HCI? (*marque sólo una opción*):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

8. ¿Ha desarrollado productos software de mercado, basado en un portafolio, aplicando técnicas de HCI? (*Conteste esta pregunta sólo si respondió "Si" en la pregunta anterior marque sólo una opción*):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------



9. ¿Se considera usted experto en el desarrollo de aplicaciones software? (*marque sólo una opción*):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

10. Durante su experiencia en el desarrollo de aplicaciones software, ¿cuáles Lenguajes de Modelado para desarrollo de aplicaciones software ha utilizado? (*Conteste esta pregunta sólo si respondió "Si" en la pregunta anterior*):

11. Durante su experiencia en el desarrollo de aplicaciones software, ¿cuáles Frameworks para desarrollo de aplicaciones software ha utilizado? (*Conteste esta pregunta sólo si se considera experto en desarrollo de aplicaciones software*):

12. Durante su experiencia en el desarrollo de aplicaciones software, ¿Cuáles de las siguientes notaciones ha utilizado?:

<input type="checkbox"/>	UML-RUP-(OMG, 2008) OpenUp (Balduino, 2007)
<input type="checkbox"/>	BPMN (OMG, 2011)
<input type="checkbox"/>	CTT (Paternò, 2004)
<input type="checkbox"/>	CIAN-CIAM (Molina et al, 2008)
<input type="checkbox"/>	TOUCHE (Penichet, 2007)
<input type="checkbox"/>	Model Driven Software Development Framework (Cubillo, 2011)
<input type="checkbox"/>	UCD with Activity Modeling (Constantine, 2006)
<input type="checkbox"/>	TaskModl DiagModl (Trætterberg, 2002)
<input type="checkbox"/>	Methodology for Developing UI to Workflow IS (Guerrero, 2010)
<input type="checkbox"/>	Wisdom (Nunes, 2001)
<input type="checkbox"/>	MPIu+a (Granollers, 2004)
<input type="checkbox"/>	HAMSTERS (Barboni et al, 2010)
<input type="checkbox"/>	Ninguna

13. ¿Qué nivel de conocimiento considera que tiene sobre Lenguajes de Modelado para el desarrollo de aplicaciones software? *(El nivel de conocimiento se refiere a si se han estudiado o se conocen los Lenguajes de Modelado. Es distinto del nivel de uso o utilización. Marque sólo una opción):*

Muy alto	<input type="radio"/>	Alto	<input type="radio"/>	Bajo	<input type="radio"/>	Ninguno	<input type="radio"/>
----------	-----------------------	------	-----------------------	------	-----------------------	---------	-----------------------

14. ¿Cuáles de las siguientes notaciones conoce o ha estudiado? *(El nivel de conocimiento se refiere a si se han estudiado o se conocen las Notaciones. Es distinto del nivel de uso o utilización.):*

<input type="checkbox"/>	UML-RUP-(OMG, 2008) OpenUp (Balduino, 2007)
<input type="checkbox"/>	BPMN (OMG, 2011)
<input type="checkbox"/>	CTT (Paternò, 2004)
<input type="checkbox"/>	CIAN-CIAM (Molina et al, 2008)
<input type="checkbox"/>	TOUCHE (Penichet, 2007)
<input type="checkbox"/>	Model Driven Software Development Framework (Cubillo, 2011)
<input type="checkbox"/>	UCD with Activity Modeling (Constantine, 2006)
<input type="checkbox"/>	TaskModl DiagModl (Trætterberg, 2002)
<input type="checkbox"/>	Methodology for Developing UI to Workflow IS (Guerrero, 2010)
<input type="checkbox"/>	Wisdom (Nunes, 2001)
<input type="checkbox"/>	MPIu+a (Granollers, 2004)
<input type="checkbox"/>	HAMSTERS (Barboni et al, 2010)
<input type="checkbox"/>	Ninguna

Agradecemos su colaboración. La información aquí recolectada será utilizada con fines netamente académicos.





ANEXO B

EVALUACIÓN DE MODELADO DE TAREAS PARA SISTEMAS INTERACTIVOS - CONTEXTO

B.1 INTRODUCCIÓN

Con este documento se pretende contextualizar a los encuestados acerca de lo que comprende la Clasificación de la Actividad para Sistemas Interactivos y la Taxonomía de la Actividad propuesta.

El contexto es necesario para comprender los conceptos relacionados con la encuesta. Dichos conceptos son indispensables para conocer la propuesta de clasificación del modelado de la actividad y el conjunto de clasificadores que la componen.

B.1.1 Glosario de términos

Awareness: Se refiere a la información que presenta el sistema acerca de lo que están haciendo los otros usuarios o el mismo sistema en tiempo real, ej: la imagen del lápiz de skype que indica que la otra persona está escribiendo. En esta propuesta se diferencia el awareness a nivel de negocio y el awareness a nivel del sistema.

Clasificación del Modelado de la Actividad: se refiere a encontrar las distintas clasificaciones de la actividad según los distintos aspectos del desarrollo de software, funcionalidad, colaboración, seguridad, desarrollo de la interfaz de usuario, etc (Giraldo, 2010).

Expresividad: Se refiere al nivel de comunicación de lo que se quiere expresar. En el contexto de una notación, por ejemplo, un elemento de modelado es expresivo si realmente comunica lo que expresa.

Feedback: Se produce feedback cuando se retroalimenta al cliente o al usuario con información del sistema, ya sea mostrando la información o indicando que se requiere alguna entrada por parte del cliente o usuario. En esta propuesta se diferencia el feedback a nivel de negocio y el feedback a nivel del sistema.

Intención: No se debe confundir la Intención con la Labor. A partir de una intención, se origina la labor que realiza. Por ejemplo, la actividad – labor “Pagar un servicio público” se origina desde la intención de “Estar al día en los pagos u obligaciones” de una persona. La labor “Abrir una cuenta bancaria”, se origina desde la intención “Ahorrar dinero”. Es decir, la intención es lo que motiva a una persona a ejecutar una Labor.

Labor: Se refiere al cómo, a las actividades, tareas que describen el aspecto de la Función.



Lenguaje de Modelado: Según Selic (Selic, 2012), un Lenguaje de Modelado, en el contexto de los sistemas de software, es un lenguaje de computador destinado a construir modelos de programas de software y los contextos en los cuales operan. La definición de un lenguaje de modelado consiste de Sintaxis Abstracta, Sintaxis Concreta y Semántica.

- *Sintaxis Abstracta:* Conjunto de conceptos/constructos del lenguaje (Ontología); y reglas para combinar los conceptos del lenguaje (reglas bien formadas).
- *Sintaxis Concreta:* Notación/Representación.
- *Semántica:* El significado de los conceptos del lenguaje.

Meta: Hace referencia al Valor que se obtiene al realizar la labor, a la motivación que impulsa su realización. Para el ejemplo de la labor "*Pagar un servicio público*", la Meta o Valor sería "*Continuar recibiendo el servicio*".

Modelado de la Actividad: es la práctica que se centra en la comprensión y representación de la naturaleza del trabajo con el fin de establecer una base para el diseño de un sistema que lo soporte (Giraldo, 2010).

Responsabilidad: En términos de describir la actividad – labor, la responsabilidad describe el quién ejecuta dicha labor. Normalmente, en la especificación de un sistema interactivo, la descripción de la labor se acompaña de la descripción de la Responsabilidad. Por ejemplo, un Rol es un conjunto de responsabilidades.

B.1.2 Taxonomía de la Actividad (TxA)

La TxA para sistemas interactivos se enfoca en identificar las categorías en las que se estructura toda la labor que se encuentra presente mientras se busca una meta de negocio (Giraldo et al, 2014). La Taxonomía define un conjunto de clasificadores que representan cada aspecto del modelado de la actividad, llamado "*Estructura Taxonómica de la Actividad*" (Tabla B.1).

- **Niveles de abstracción:** La Estructura Taxonómica de la Actividad propuesta se divide en dos niveles de abstracción: Negocio y Sistema. El Nivel de Negocio se enfoca en capturar la esencia del sistema. Es un sistema libre de tecnología, en el que interesan las intenciones de los usuarios y el propósito fundamental del negocio. El Nivel de Sistema informático se centra en la definición de las actividades que están relacionadas con la interacción con el sistema.
- **Función:** Cada nivel de abstracción se clasifica en dos niveles de jerarquía entre las actividades: actividades de bajo nivel de granularidad y actividades del alto nivel de granularidad. La clasificación por tipo de actividad intenta diferenciar la información que es esencial para el diseño de la interfaz de usuario y separarla de cualquier otra información considerada útil para otros

finés. Las actividades se clasifican según su tipo en: 1) Actividades de interacción con los usuarios: actividades base y de soporte, y 2) Actividades sin interacción con el usuario: actividades de gestión.

Tabla B.1. Estructura taxonómica de la actividad

Estructura Taxonómica de la Actividad, una sola Capa													
Negocio	Granularidad	Cómo					Dato	Meta		Quién		Ubicación	Tiempo
		Base		Soporte		Gestión		O	Re	Cl	Wo		
		Tc	I-A	Tc	I-A	Tw							
Sistema	Granularidad Baja												
	Granularidad Alta												
		Cómo					Dato	Meta		Quién		Ubicación	Tiempo
		Base		Soporte		Gestión		O	Re	Us	Si		
	Tu	I	Tu	I	Ts								
	Granularidad Baja												
	Granularidad Alta												

Tc: Tarea que hace el Cliente
 I-A: Inter-Acción
 Tw: Tarea que hace el Worker
 O: Objetivo
 Re: Regla
 Cl: Cliente
 Wo: Trabajador (Worker)
 Tu: Tarea que hace el Usuario
 I: Interacción
 Ts: Tarea que hace el Sistema
 Us: Usuario
 Si: Sistema

- **Datos** (qué): Modelan los conceptos o cosas que son importantes para la ejecución de la actividad y por tanto que manipula.
- **Meta** a motivación (Por qué). Supone la representación descriptiva de la motivación que impulsa la ejecución de la actividad.
- **Quién**: Las personas. Especifica los responsables de ejecutar las actividades.
- **Ubicación**: La red (dónde). Describe elementos de la arquitectura donde se lleva a cabo la actividad. Localidades, componentes, computadores, etc.
- **Tiempo** (cuándo): Establece los calendarios del sistema. Describe eventos que disparan la ejecución de las actividades, periodicidad, sincronización, etc.

B.1.3 Enunciado del caso de estudio

La encuesta se estructura a partir del caso de estudio que describe la actividad "Retirar Dinero". Se describen los pasos que componen la actividad para luego analizar cómo se clasifica la información capturada por la especificación.



Escenario de un retiro de dinero por ventanilla en una entidad bancaria

Pasos que componen la actividad:

1. El Cliente ingresa a la entidad bancaria y busca un formato de "Transacciones".
2. El Cliente diligencia la información solicitada en el formato "Transacciones": "Tipo de Cuenta", "Número de Cuenta", "Cantidad a Retirar", "Nombre, Identificación, Teléfono", de quien realiza la transacción.
3. El Cliente decide en qué fila ubicarse (Fila de Cliente Preferencial, Fila de Cliente No Preferencial).
4. El Cliente espera su turno para que lo atienda el Señor Cajero.
5. Cuando llega su turno, el Cliente se dirige al Cajero y le entrega el formato de "Transacciones" diligenciado.
6. El Cajero verifica la información de la cuenta bancaria diligenciada por el Cliente en el formato de "Transacciones":
 - a. El Cajero decide qué tipo de transacción va a realizar en el sistema.
 - b. El Cajero selecciona la opción "Retiro de Dinero". (Se asume que el Cajero previamente ha ingresado al sistema y se ha autenticado).
 - c. El Sistema solicita la información del cliente y del monto a retirar. (Si el monto es inferior a la suma permitida para retiro a través de cajero automático, se cobra una comisión adicional al Cliente).
 - d. El Cajero ingresa la información solicitada.
 - e. El Sistema valida la información (Si existe la cuenta, si existe el cliente).
 - f. El Sistema informa al Cajero el resultado de la validación. (Se asume que sí existe la cuenta y el cliente).
7. El Cajero solicita al Cliente que inserte su tarjeta en el Datáfono.
8. El Cliente inserta su tarjeta en el Datáfono.
9. El Sistema lee la tarjeta.
10. El Sistema valida la información de la tarjeta.
11. El Sistema informa al Cajero que la tarjeta es válida.
12. El Sistema indica al Cliente que retire la tarjeta. El Cliente retira la tarjeta. (Esta es una medida de seguridad para asegurar que el Usuario no deje su tarjeta en la máquina).
13. El Sistema pide al Cliente que digite una clave. El Cliente digita el PIN.
14. El Sistema valida el PIN.
15. El Sistema informa al Cajero y al Cliente que el PIN es válido.
16. El Sistema muestra al Cajero la información de la cuenta bancaria.
17. El Cajero chequea si la cuenta bancaria tiene la cantidad requerida por el Cliente.
18. El Cajero ingresa al sistema la información sobre la cantidad de dinero a retirar.
19. El Sistema descuenta la cantidad requerida de la cuenta bancaria y compromete la transacción.
20. El Sistema envía un mensaje de texto informando al Cliente sobre la transacción realizada y el nuevo saldo disponible.

21. El Cajero cuenta la cantidad de dinero solicitado por el cliente.
22. El Cajero pone un sello al formato de "Transacciones" (original y copia) con la fecha de la transacción.
23. El Cajero guarda el original del formato de "Transacciones".
24. El Cajero entrega el dinero al Cliente junto con una copia del formato de "Transacciones".
25. El Cliente guarda el dinero y el formato y se retira de la casilla.
26. La Actividad "Retiro de Dinero" finaliza.

B.1.4 Identificación de Elementos de la Taxonomía de la Actividad, de acuerdo al enunciado del Caso de Estudio

En primer lugar, se clasifica la Actividad como de *Tipo Base*, por ser una Actividad que tiene interacción con los usuarios y por ser una de las Actividades principales que se realiza en una Entidad Bancaria.

De acuerdo a la TxA propuesta, la sección B.1.4.1, detalla cada tarea y los clasificadores correspondientes:

- Tc: Tarea que hace el Cliente
- I-A: Inter-Acción
- Tw: Tarea que hace el Worker
- Tu: Tarea que hace el Usuario
- I: Interacción
- Ts: Tarea que hace el Sistema
- Dato (Negocio y Sistema)
- Meta (Negocio y Sistema)
- Objetivo (Negocio y Sistema)
- Regla (Negocio y Sistema)
- Cliente
- Worker
- Usuario
- Sistema
- Ubicación (Negocio y Sistema)
- Tiempo (Negocio y Sistema)

Adicionalmente, se tendrán en cuenta los clasificadores "Feedback" y "Awareness" (ambos, de negocio y de sistema), identificados como sub-clasificadores de Inter-Acción e Interacción, respectivamente.

Posteriormente, cada enunciado es ubicado en las tablas B.2 y B.3, de acuerdo al nivel de abstracción (Negocio y Sistema).



B.1.4.1 Escenario de un retiro de dinero por ventanilla en una entidad bancaria, clasificado con la TxA:

En esta sección, por cada paso se identifican los clasificadores que componen cada tarea, de acuerdo a la Estructura de la Taxonomía de la Actividad.

Nótese que por cada tarea, puede ser que se identifiquen clasificadores que no necesariamente están especificados en la redacción de la tarea, pero que luego serán necesarios en la especificación si se desea mejorar la interacción de los usuarios con el sistema en desarrollo.

Pasos que componen la actividad y clasificadores asociados:

1. El Cliente ingresa a la entidad bancaria y busca un formato de "Transacciones".

- El Cliente: <Cliente>.
- Entidad Bancaria: <Ubicación de negocio>.
- El Cliente ingresa a la entidad bancaria: <Inter-Acción>.
- El Cliente decide qué formato buscar: <Tarea que hace el Cliente>.
- El Cliente busca un formato de "Transacciones": <Inter-Acción>.
- Formato de Transacciones: <Datos de negocio>.

2. El Cliente diligencia la información solicitada en el formato "Transacciones": "Tipo de Cuenta", "Número de Cuenta", "Cantidad a Retirar", "Nombre, Identificación, Teléfono", de quien realiza la transacción.

- El Cliente: <Cliente>.
- El Cliente diligencia la información solicitada en el formato "Transacciones": <Inter-Acción>.
- "Tipo de Cuenta", "Número de Cuenta", "Cantidad a Retirar", "Nombre, Identificación, Teléfono": <Datos de negocio>.
- Diligenciar correctamente el Formulario: <Objetivo de Negocio>.

3. El Cliente decide en qué fila ubicarse (Fila de Cliente Preferencial, Fila de Cliente No Preferencial).

- El Cliente decide en qué fila ubicarse (Fila de Cliente Preferencial, Fila de Cliente No Preferencial): <Tarea que hace el cliente>.
- Fila de Cliente Preferencial, Fila de Cliente No Preferencial: <Datos de Negocio>.
- Hacer fila: <Regla de negocio>.

4. El Cliente espera su turno para que lo atienda el Señor Cajero.

- El Cliente espera su turno para que lo atienda el Señor Cajero: <Tarea que hace el Cliente>.
- Esperar Turno: <Tiempo>.
- Señor Cajero: <Worker>.

5. Cuando llega su turno, el Cliente se dirige al Cajero y le entrega el formato de "Transacciones" diligenciado.

- Cuando llega su turno, el Cliente se dirige al Cajero: <Tarea que hace el Cliente>.

- El Cliente entrega al Cajero el formato de "Transacciones" diligenciado: *<Inter-Acción>*.

6. El Cajero verifica la información de la cuenta bancaria diligenciada por el Cliente en el formato de "Transacciones":

- El Cajero verifica la información de la cuenta bancaria diligenciada por el Cliente en el formato de "Transacciones": *<Tarea que hace el Worker>*.
- El Cajero está verificando la información: *<Awareness de Negocio>*.

o El Cajero decide qué tipo de transacción va a realizar en el sistema.

El Cajero decide qué tipo de transacción va a realizar en el sistema: *<Tarea que hace el Usuario>*.

Cajero: *<Usuario>*.

Transacción: *<Dato de Sistema>*.

o El Cajero selecciona la opción "Retiro de Dinero". (Se asume que el Cajero previamente ha ingresado al sistema y se ha autenticado).

El Cajero selecciona la opción "Retiro de Dinero": *<Interacción>*.

o El Sistema solicita la información del cliente y del monto a retirar. (Si el monto es inferior a la suma permitida para retiro a través de cajero automático, se cobra una comisión adicional al Cliente).

El Sistema solicita la información del cliente y del monto a retirar: *<Interacción>*.

Información del Cliente: *<Datos de Sistema>*.

Si el monto es inferior a la suma permitida para retiro a través de cajero automático, se cobra una comisión adicional al Cliente: *<Regla de Sistema>*.

o El Cajero ingresa la información solicitada.

El Cajero ingresa la información solicitada: *<Interacción>*.

o El Sistema valida la información (Si existe la cuenta, si existe el cliente).

El Sistema valida la información (Si existe la cuenta, si existe el cliente): *<Tarea que hace el Sistema>*.

Sistema: *<Sistema>*.

Validando información: *<Interacción - Awareness de Sistema>*.

o El Sistema informa al Cajero el resultado de la validación. (Se asume que sí existe la cuenta y el cliente).

El Sistema informa al Cajero el resultado de la validación. (Se asume que sí existe la cuenta y el cliente): *<Interacción - Feedback de Sistema>*.

Cuenta, Cliente: *<Datos de sistema>*.

7. El Cajero solicita al Cliente que inserte su tarjeta en el Datáfono.

- El Cajero solicita al Cliente que inserte su tarjeta en el Datáfono: *<Inter-Acción>*.

8. El Cliente inserta su tarjeta en el Datáfono.

- El Cliente inserta su tarjeta en el Datáfono: *<Interacción>*.



9. El Sistema la lee la tarjeta.

- El Sistema la lee la tarjeta: *<Tarea que hace el sistema>*.
- El Sistema está leyendo la tarjeta: *<Interacción - Awareness de Sistema>*.

10. El Sistema valida la información de la tarjeta.

- El Sistema valida la información de la tarjeta: *<Tarea que hace el sistema>*.
- Validando información: *<Interacción – Awareness de Sistema>*.
- Información de la Tarjeta *<Datos de sistema>*.

11. El Sistema informa al Cajero que la tarjeta es válida.

- El Sistema informa al Cajero que la tarjeta es válida: *<Interacción - Feedback de Sistema>*.
- Cajero: *<Usuario>*.

12. El Sistema indica al Cliente que retire la tarjeta. El Cliente retira la tarjeta. (Esta es una medida de seguridad para asegurar que el Usuario no deje su tarjeta en la máquina).

- El Sistema indica al Cliente que retire la tarjeta: *<Interacción>*.
- El Cliente retira la tarjeta: *<Interacción>*.
- El Sistema indica que ya no está leyendo la tarjeta: *<Interacción - Feedback de Sistema>*.

13. El Sistema pide al Cliente que digite una clave. El Cliente digita la clave.

- El Sistema pide al Cliente que digite una clave: *<Interacción>*.
- El Cliente digita la clave: *<Interacción>*.
- Clave: *<Dato de Sistema>*.
- El Sistema queda en espera a que el Cliente digite la clave: *<Tiempo>*.

14. El Sistema valida la clave.

- El Sistema valida la clave: *<Tarea que hace el Sistema>*.
- Validando clave: *<Interacción - Awareness de Sistema>*.
- El Sistema toma un tiempo en validar la clave: *<Tiempo>*.

15. El Sistema informa al Cajero y al Cliente que la clave es válida.

- El Sistema informa al Cajero y al Cliente que la clave es válida: *<Interacción – Feedback de Sistema>*.

16. El Sistema muestra al Cajero la información de la cuenta bancaria.

- El Sistema muestra al Cajero la información de la cuenta bancaria: *<Interacción>*.
- Información de la Cuenta Bancaria: *<Datos de Sistema>*.

17. El Cajero chequea si la cuenta bancaria tiene la cantidad requerida por el Cliente.

- El Cajero chequea si la cuenta bancaria tiene la cantidad requerida por el Cliente: *<Tarea que hace el usuario>*.
- El Cliente observa que el cajero está chequeando su saldo disponible: *<Inter-Acción, - Awareness de Negocio>*.

18. El Cajero ingresa al sistema la información sobre la cantidad de dinero a retirar.

- El Cajero ingresa al sistema la información sobre la cantidad de dinero a retirar: *<Interacción>*.
- El Cajero tarda un tiempo en ingresar la información: *<Tiempo>*.
- El Cliente espera mientras el Cajero realiza la transacción: *<Inter-Acción, - Awareness de Negocio>*.

19. El Sistema descuenta la cantidad requerida de la cuenta bancaria y compromete la transacción.

- El Sistema descuenta la cantidad requerida de la cuenta bancaria y compromete la transacción: *<Tarea que hace el sistema>*.
- El Sistema tarda un tiempo en realizar la transacción: *<Tiempo>*.
- El Sistema debe informar al Usuario (Cajero) que está realizando la transacción: *<Interacción, - Awareness de Sistema>*.
- El Sistema debe informar al Usuario (Cajero) que comprometió la transacción: *<Interacción – Feedback de Sistema>*.
- Cantidad retirada de la cuenta: *<Datos de Sistema>*.
- La Transacción queda comprometida en el Sistema Local y en el Sistema Central: *<Objetivo de Sistema>*.
- Sistema Central y Sistema Local: *< Ubicación de Sistema >*

20. El Sistema envía un mensaje de texto informando al Cliente sobre la transacción realizada y el nuevo saldo disponible.

- El Sistema envía un mensaje de texto informando al Cliente sobre la transacción realizada y el nuevo saldo disponible: *<Tarea que hace el sistema>*.
- El Cliente recibe una alerta de mensaje de texto en su teléfono móvil: *<Interacción – Feedback de Sistema>*.

21. El Cajero cuenta la cantidad de dinero solicitado por el cliente.

- El Cajero cuenta la cantidad de dinero solicitado por el cliente: *<Tarea que hace el Worker>*.
- El Cajero tarda un tiempo contando el dinero: *<Tiempo>*.
- El Cliente espera mientras el Cajero cuenta el dinero: *<Inter-Acción, - Awareness de Negocio>*.
- Dinero: *<Datos de Negocio>*.

22. El Cajero pone un sello al formato de "Transacciones" (original y copia) con la fecha de la transacción.

- El Cajero pone un sello al formato de "Transacciones" (original y copia) con la fecha de la transacción: *<Tarea que hace el Worker>*.
- Formato de Transacciones: *<Datos de Negocio>*.
- Fecha de la Transacción: *<Datos de Negocio>*.
- El Cliente chequea que el Cajero está sellando el formato de transacciones: *<Inter-Acción, - Awareness de Negocio>*.

23. El Cajero guarda el original del formato de "Transacciones".

- El Cajero guarda el original del formato de "Transacciones": *<Tarea que hace el Worker>*.
- Lugar en el que se guarda el original del formato: *<Ubicación de Negocio>*.

24. El Cajero entrega el dinero al Cliente junto con una copia del formato de "Transacciones".

- El Cajero entrega el dinero al Cliente junto con una copia del formato de "Transacciones": *<Inter-Acción>*.

25. El Cliente guarda el dinero y el formato y se retira de la casilla.

- El Cliente guarda el dinero y el formato y se retira de la casilla: *<Tarea que hace el Cliente>*.

26. La Actividad "Retiro de Dinero" finaliza.



Tabla B.2a. Clasificación de la actividad para la actividad de tipo base: retirar dinero (nivel de abstracción negocio)

Estructura Taxonomica de la Actividad, (Negocio)											
Cómo		Gestión		Dato		Meta		Quién		Ubicación	Tiempo
Base	I-A	T _w			O	Re	CI	Vigilante	Zona de ingreso	Horario de atención	
1.1.El Cliente decide que formato buscar	1.1.El Cliente ingresa a la entidad bancaria		Información del cliente	Autenticarse	Cliente		Cliente				
1.2. El Cliente decide que formato buscar	1.3. El Cliente busca un Formato de Transacciones		Formato de Transacciones		Cliente		Cliente				
	2.El Cliente diligencia la información solicitada en el formato "Transacciones"		Tipo de Cuenta", "Número de Cuenta", "Cantidad a Retirar", "Nombre, Identificación, Teléfono"	Diligenciar correctamente el formulario	Cliente		Cliente				
3.El Cliente decide en qué fila ubicarse			Fila de Cliente Preferencial, Fila de Cliente No Preferencial	Hacer Fila							
4.El Cliente espera su turno para que lo atienda el Señor Cajero								Cajero		Esperar Turno	
5.Cuando llega su turno, el Cliente se dirige al Cajero	5.1. El Cliente le entrega al Cajero el formato de "Transacciones" diligenciado		Formato de Transacciones		Cliente		Cliente	Cajero			
	6.El Cajero verifica la información de la cuenta bancaria diligenciada por el Cliente en el formato de "Transacciones"										
	6.1. El Cajero está verificando la información: <Avarness de Negocio>										
	13.El Cajero solicita al Cliente que inserte su tarjeta en el Datafono.										
	26.1. El Cliente observa que el										



Tabla B.2b. Clasificación de la actividad para la actividad de tipo base: retirar dinero (nivel de abstracción negocio)

	carero está chequeando su saldo disponible: <Awareness de Negocio>																					
	27.1. El Cliente espera mientras el Cajero realiza la transacción: <Awareness de Negocio>																					
	30. El Cajero cuenta la cantidad de dinero solicitado por el cliente.																				El Cajero tarda un tiempo contando el dinero	
	30.1. El Cliente espera mientras el Cajero cuenta el dinero: <Awareness de Negocio>						Dinero															
	31. El Cajero pone un sello al formato de "Transacciones" (original y copia) con la fecha de la transacción.																					
	31.1. El Cliente chequea que el Cajero está sellando el formato de transacciones: <Awareness de Negocio>																					
	32. El Cajero guarda el original del formato de "Transacciones".																					Lugar en el que se guarda el original del formato
	33. El Cajero entrega el dinero al Cliente junto con una copia del formato de "Transacciones".																					
	34. El Cliente guarda el dinero y el formato y se retira de la casilla.																					

Tabla B.3a. Clasificación de la actividad para la actividad de tipo base: retirar dinero (nivel de abstracción sistema)

Estructura Taxonomica de la Actividad. (Sistema)										
Base	Cómo		Gestión	Dato	Meta		Quién		Ubicación	Tiempo
	Tu	I			O	Re	Us	Si		
7.El Cajero decide que tipo de transacción va a realizar en el sistema				Transacción			Cajero			Tiempo que toma la decisión
8. El Cajero selecciona la opción "Retiro de Dinero".							Cajero			
9. El Sistema solicita la información del cliente y del monto a retirar.				Información del Cliente		Si el monto es inferior a la suma permitida para retiro a través de cajero automático, se cobra una comisión adicional al Cliente		Sistema		
10. El Cajero ingresa la información solicitada.							Cajero			
11.El Sistema valida la información								Sistema		
El Sistema está validando la información: <Avariness de Sistema>										
12.El Sistema informa al Cajero el resultado de la validación: <Feedback de Sistema>				Cuenta, Cliente				Sistema		
14.El Cliente busca y encuentra su tarjeta del banco										
15. El Cliente inserta su tarjeta en el Datafono.							Cliente			
16. El Sistema lee la								Sistema		





ANEXO C

SELECCIÓN DE ELEMENTOS DE MODELADO DE SISTEMAS INTERACTIVOS

C.1 NIVEL DE NEGOCIO

Con esta encuesta se pretende seleccionar un conjunto de elementos notacionales para representar los clasificadores definidos en la Taxonomía de la Actividad para Sistemas Interactivos.

Dinámica para contestar la encuesta

La dinámica que se seguirá con esta encuesta es: Se presenta cada clasificador definido en la Taxonomía de la Actividad (TxA) y un fragmento de la especificación para la Actividad "Retirar Dinero" que relacione dicho clasificador. El encuestado analiza y contesta las preguntas de acuerdo a lo que se plantea en el fragmento de la especificación. Se debe consultar simultáneamente el documento del Contexto.

1. Nombre del encuestado: _____

La TxA define el clasificador <Tarea que hace el Cliente>, en el nivel de Negocio (Nivel independiente de la tecnología). Ejemplo: "El Cliente decide qué formato buscar": <Tarea que hace el Cliente>.

Ahora, conteste las preguntas 1 a 10 de acuerdo a dicha especificación.

1. ¿Ha usado el clasificador <Tarea que hace el Cliente> en sus proyectos? (*marque sólo una opción*):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

2. ¿Ha estudiado el clasificador <Tarea que hace el Cliente> en sus lecturas? (*marque sólo una opción*):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

De acuerdo a la figura C.1 conteste las preguntas 3, 4, 5, 6, 7

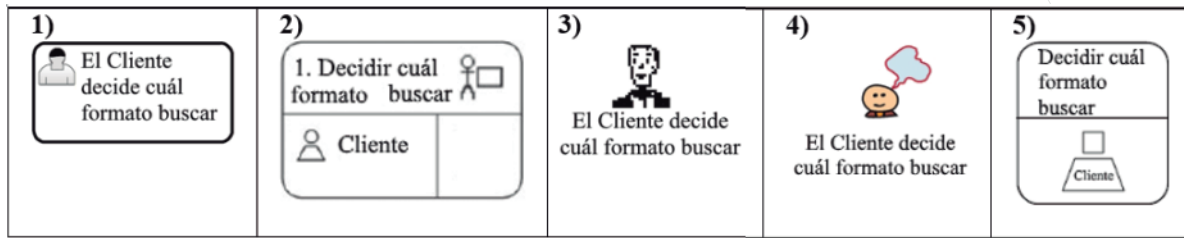


Figura C.1. Símbolos de algunas propuestas existentes para representar - Tarea que hace el Cliente.

3. De los símbolos en la figura C.1 "Tarea que hace el Cliente", (marque sólo una opción):

Conoce 1	<input type="radio"/>	Conoce más de 1	<input type="radio"/>	Conoce ninguno	<input type="radio"/>
----------	-----------------------	-----------------	-----------------------	----------------	-----------------------

4. ¿Ha usado alguno de estos símbolos en sus proyectos?(marque sólo una opción):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

5. ¿Ha estudiado alguno de estos símbolos en sus lecturas?(marque sólo una opción):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

6. Según mi experiencia de trabajar en proyectos con esa notación, reconozco al número ___ que sería la mejor opción para representar <Tarea que hace el Cliente> (rellene con el número 0 para "ninguno").

7. Por su significado, expresividad y lo que me da a entender, me parece que es mejor el número ___ para representar <Tarea que hace el Cliente> (rellene con el número 0 para "ninguno").

8. ¿Ha visto otro símbolo que le parezca mejor para representar <Tarea que hace el Cliente>?(marque sólo una opción):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

Si su respuesta fue afirmativa responda la pregunta 9, de lo contrario continúe con la pregunta 10.

9. ¿Cuál es el nombre de la propuesta donde encontró el símbolo?:

10. ¿Le gustaría sugerir un símbolo para representar el clasificador <Tarea que hace el Cliente>?(marque sólo una opción):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

Si su respuesta fue afirmativa, envíe su propuesta de símbolo al siguiente correo: mlvillegas@uniquindio.edu.co.

La TxA define el clasificador <Cliente>, en el nivel de Negocio (Nivel independiente de la tecnología). Ejemplo: "El Cliente": <Cliente>.

Ahora, conteste las preguntas 1 a 10 de acuerdo a dicha especificación.

1. ¿Ha usado el clasificador <Cliente> en sus proyectos? (marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

2. ¿Ha estudiado el clasificador <Cliente> en sus lecturas? (marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

De acuerdo a la figura C.2 conteste las preguntas 3, 4, 5, 6, 7

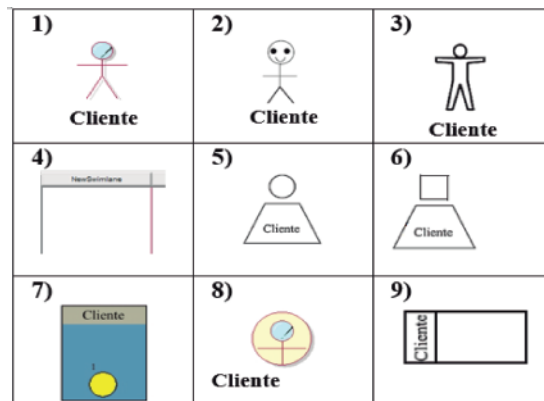


Figura C.2. Símbolos de algunas propuestas existentes para representar - Cliente.

3. De los símbolos en la figura C.2 "Cliente", (marque sólo una opción):

Conoce 1 <input type="radio"/>	Conoce más de 1 <input type="radio"/>	Conoce ninguno <input type="radio"/>
--------------------------------	---------------------------------------	--------------------------------------

4. ¿Ha usado alguno de estos símbolos en sus proyectos?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

5. ¿Ha estudiado alguno de estos símbolos en sus lecturas?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

6. Según mi experiencia de trabajar en proyectos con esa notación, reconozco al



número ____ que sería la mejor opción para representar <Cliente> (rellene con el número 0 para “ninguno”).

7. Por su significado, expresividad y lo que me da a entender, me parece que es mejor el número ____ para representar <Cliente> (rellene con el número 0 para “ninguno”).

8. ¿Ha visto otro símbolo que le parezca mejor para representar <Cliente>?(marque sólo una opción):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

Si su respuesta fue afirmativa responda la pregunta 9, de lo contrario continúe con la pregunta 10.

9. ¿Cuál es el nombre de la propuesta donde encontró el símbolo?:

10. ¿Le gustaría sugerir un símbolo para representar el clasificador <Cliente>?(marque sólo una opción):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

Si su respuesta fue afirmativa, envíe su propuesta de símbolo al siguiente correo: mlvillegas@uniquindio.edu.co.

La TxA define el clasificador <Ubicación de Negocio>, en el nivel de Negocio (Nivel independiente de la tecnología). Ejemplo: "Entidad Bancaria": <Ubicación de Negocio>.

Ahora, conteste las preguntas 1 a 10 de acuerdo a dicha especificación.

1. ¿Ha usado el clasificador <Ubicación de Negocio> en sus proyectos? (marque sólo una opción):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

2. ¿Ha estudiado el clasificador <Ubicación de Negocio> en sus lecturas? (marque sólo una opción):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

De acuerdo a la figura C.3 conteste las preguntas 3, 4, 5, 6, 7



Figura C.3. Símbolos de algunas propuestas existentes para representar - Ubicación de Negocio.

3. De los símbolos en la figura C.3 "*Ubicación de Negocio*", (*marque sólo una opción*):

Conoce 1 <input type="radio"/>	Conoce más de 1 <input type="radio"/>	Conoce ninguno <input type="radio"/>
--------------------------------	---------------------------------------	--------------------------------------

4. ¿Ha usado alguno de estos símbolos en sus proyectos?(*marque sólo una opción*):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

5. ¿Ha estudiado alguno de estos símbolos en sus lecturas?(*marque sólo una opción*):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

6. Según mi experiencia de trabajar en proyectos con esa notación, reconozco al número ___ que sería la mejor opción para representar <*Ubicación de Negocio*> (*rellene con el número 0 para "ninguno"*).

7. Por su significado, expresividad y lo que me da a entender, me parece que es mejor el número ___ para representar <*Ubicación de Negocio*> (*rellene con el número 0 para "ninguno"*).

8. ¿Ha visto otro símbolo que le parezca mejor para representar <*Ubicación de Negocio*>?(*marque sólo una opción*):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

Si su respuesta fue afirmativa responda la pregunta 9, de lo contrario continúe con la pregunta 10.

9. ¿Cuál es el nombre de la propuesta donde encontró el símbolo?:

10. ¿Le gustaría sugerir un símbolo para representar el clasificador <*Ubicación de Negocio*>?(*marque sólo una opción*):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------



Si su respuesta fue afirmativa, envíe su propuesta de símbolo al siguiente correo: mlvillegas@uniquindio.edu.co.

La TxA define el clasificador <Dato de Negocio>, en el nivel de Negocio (Nivel independiente de la tecnología). Ejemplo: "Cuenta Bancaria": <Dato de Negocio>.

Ahora, conteste las preguntas 1 a 10 de acuerdo a dicha especificación.

1. ¿Ha usado el clasificador <Dato de Negocio> en sus proyectos? (marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

2. ¿Ha estudiado el clasificador <Dato de Negocio> en sus lecturas? (marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

De acuerdo a la figura C.4 conteste las preguntas 3, 4, 5, 6, 7

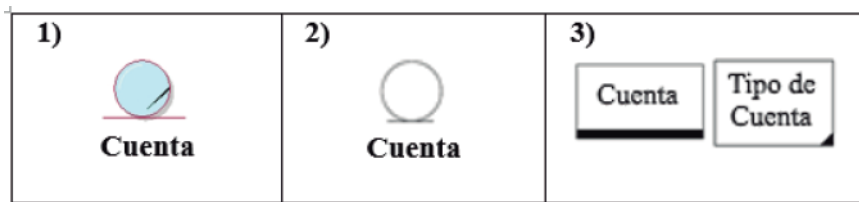


Figura C.4. Símbolos de algunas propuestas existentes para representar - Dato de Negocio.

3. De los símbolos en la figura C.4 "Dato de Negocio", (marque sólo una opción):

Conoce 1 <input type="radio"/>	Conoce más de 1 <input type="radio"/>	Conoce ninguno <input type="radio"/>
--------------------------------	---------------------------------------	--------------------------------------

4. ¿Ha usado alguno de estos símbolos en sus proyectos?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

5. ¿Ha estudiado alguno de estos símbolos en sus lecturas?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

6. Según mi experiencia de trabajar en proyectos con esa notación, reconozco al número ____ que sería la mejor opción para representar <Dato de Negocio> (rellene con el número 0 para "ninguno").

7. Por su significado, expresividad y lo que me da a entender, me parece que es mejor el número ____ para representar <Dato de Negocio> (rellene con el número 0 para "ninguno").

8. ¿Ha visto otro símbolo que le parezca mejor para representar <Dato de Negocio>?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

Si su respuesta fue afirmativa responda la pregunta 9, de lo contrario continúe con la pregunta 10.

9. ¿Cuál es el nombre de la propuesta donde encontró el símbolo?:

10. ¿Le gustaría sugerir un símbolo para representar el clasificador <Dato de Negocio>?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

Si su respuesta fue afirmativa, envíe su propuesta de símbolo al siguiente correo: mlvillegas@uniquindio.edu.co.

La TxA define el clasificador <Inter-Acción>, en el nivel de Negocio (Nivel independiente de la tecnología). Ejemplo: "El Cliente diligencia la información solicitada en el formato "Transacciones": <Inter-Acción>.

Ahora, conteste las preguntas 1 a 10 de acuerdo a dicha especificación.

1. ¿Ha usado el clasificador <Inter-Acción> en sus proyectos? (marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

2. ¿Ha estudiado el clasificador <Inter-Acción> en sus lecturas? (marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

De acuerdo a la figura C.5 conteste las preguntas 3, 4, 5, 6, 7

3. De los símbolos en la figura C.5 "Inter-Acción", (marque sólo una opción):

Conoce 1 <input type="radio"/>	Conoce más de 1 <input type="radio"/>	Conoce ninguno <input type="radio"/>
--------------------------------	---------------------------------------	--------------------------------------

4. ¿Ha usado alguno de estos símbolos en sus proyectos?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

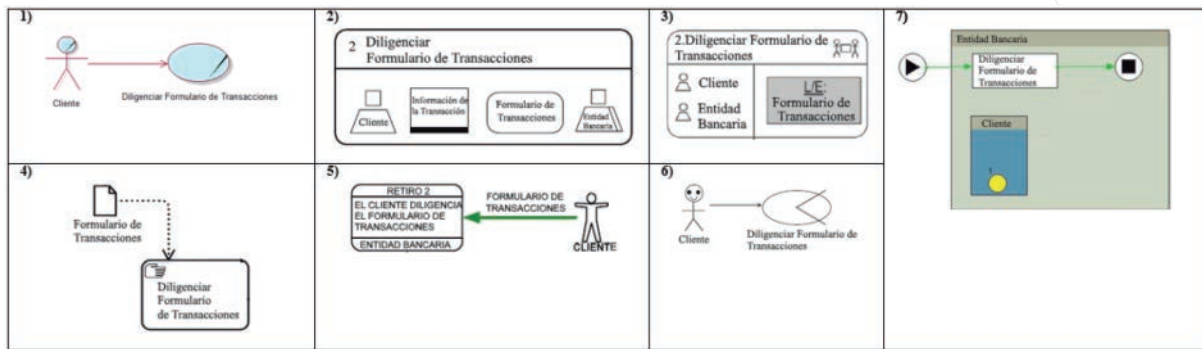


Figura C.5. Símbolos de algunas propuestas existentes para representar - Inter-Acción

5. ¿Ha estudiado alguno de estos símbolos en sus lecturas?(marque sólo una opción):

Si No

6. Según mi experiencia de trabajar en proyectos con esa notación, reconozco al número ____ que sería la mejor opción para representar <Inter-Acción> (rellene con el número 0 para "ninguno").

7. Por su significado, expresividad y lo que me da a entender, me parece que es mejor el número ____ para representar <Inter-Acción> (rellene con el número 0 para "ninguno").

8. ¿Ha visto otro símbolo que le parezca mejor para representar <Inter-Acción>?(marque sólo una opción):

Si No

Si su respuesta fue afirmativa responda la pregunta 9, de lo contrario continúe con la pregunta 10.

9. ¿Cuál es el nombre de la propuesta donde encontró el símbolo?:

10. ¿Le gustaría sugerir un símbolo para representar el clasificador <Inter-Acción>?(marque sólo una opción):

Si No

Si su respuesta fue afirmativa, envíe su propuesta de símbolo al siguiente correo: mlvillegas@uniquindio.edu.co.

La TxA define el clasificador <Objetivo de Negocio>, en el nivel de Negocio (Nivel independiente de la tecnología). Ejemplo: "Formulario correctamente diligenciado": <Objetivo de Negocio>.

Ahora, conteste las preguntas 1 a 10 de acuerdo a dicha especificación.

1. ¿Ha usado el clasificador <Objetivo de Negocio> en sus proyectos? (marque sólo una opción):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

2. ¿Ha estudiado el clasificador <Objetivo de Negocio> en sus lecturas? (marque sólo una opción):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

De acuerdo a la figura C.6 conteste las preguntas 3, 4, 5, 6, 7

Figura C.5. Símbolos de algunas propuestas existentes para representar - Objetivo de Negocio.



3. Conoce el símbolo de la figura C.6 "Objetivo de Negocio", (marque sólo una opción):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

4. ¿Ha usado este símbolo en sus proyectos?(marque sólo una opción):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

5. ¿Ha estudiado este símbolo en sus lecturas?(marque sólo una opción):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

6. Según mi experiencia de trabajar en proyectos con esa notación, reconozco al número ___ que sería la mejor opción para representar <Objetivo de Negocio> (rellene con el número 0 para "ninguno").

7. Por su significado, expresividad y lo que me da a entender, me parece que es mejor el número ___ para representar <Objetivo de Negocio> (rellene con el número 0 para "ninguno").



8. ¿Ha visto otro símbolo que le parezca mejor para representar <Objetivo de Negocio>?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

Si su respuesta fue afirmativa responda la pregunta 9, de lo contrario continúe con la pregunta 10.

9. ¿Cuál es el nombre de la propuesta donde encontró el símbolo?:

10. ¿Le gustaría sugerir un símbolo para representar el clasificador <Objetivo de Negocio>?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

Si su respuesta fue afirmativa, envíe su propuesta de símbolo al siguiente correo: mlvillegas@uniquindio.edu.co.

La TxA define el clasificador <Regla de Negocio>, en el nivel de Negocio (Nivel independiente de la tecnología). Ejemplo: "Hacer fila": <Regla de Negocio>.

Ahora, conteste las preguntas 1 a 10 de acuerdo a dicha especificación.

1. ¿Ha usado el clasificador <Regla de Negocio> en sus proyectos? (marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

2. ¿Ha estudiado el clasificador <Regla de Negocio> en sus lecturas? (marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

De acuerdo a la figura C.7 conteste las preguntas 3, 4, 5, 6, 7



Figura C.7. Símbolos de algunas propuestas existentes para representar - Regla de Negocio.

3. Conoce el símbolo de la figura C.7 "Regla de Negocio", (marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

4. ¿Ha usado este símbolo en sus proyectos?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

5. ¿Ha estudiado este símbolo en sus lecturas?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

6. Según mi experiencia de trabajar en proyectos con esa notación, reconozco al número ____ que sería la mejor opción para representar <Regla de Negocio> (rellene con el número 0 para "ninguno").

7. Por su significado, expresividad y lo que me da a entender, me parece que es mejor el número ____ para representar <Regla de Negocio> (rellene con el número 0 para "ninguno").

8. ¿Ha visto otro símbolo que le parezca mejor para representar <Regla de Negocio>?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

Si su respuesta fue afirmativa responda la pregunta 9, de lo contrario continúe con la pregunta 10.

9. ¿Cuál es el nombre de la propuesta donde encontró el símbolo?:

10. ¿Le gustaría sugerir un símbolo para representar el clasificador <Regla de Negocio>?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

Si su respuesta fue afirmativa, envíe su propuesta de símbolo al siguiente correo: mlvillegas@uniquindio.edu.co.

La TxA define el clasificador <Worker>, en el nivel de Negocio (Nivel independiente de la tecnología). Ejemplo: "El Cajero": <Worker>.

Ahora, conteste las preguntas 1 a 10 de acuerdo a dicha especificación.

1. ¿Ha usado el clasificador <Worker> en sus proyectos? (marque sólo una opción):

2. ¿Ha estudiado el clasificador <Worker> en sus lecturas? (marque sólo una opción):

De acuerdo a la figura C.8 conteste las preguntas 3, 4, 5, 6, 7

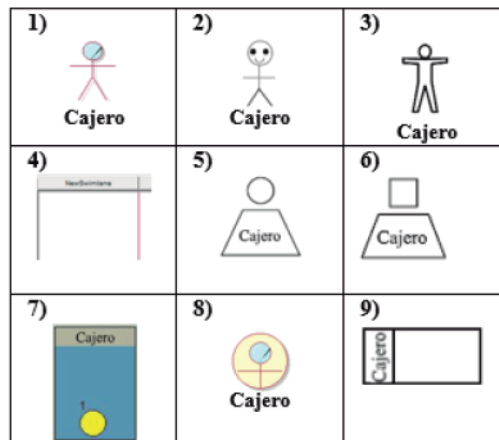


Figura C.8. Símbolos de algunas propuestas existentes para representar - Worker

3. De los símbolos en la figura C.8 "Worker", (marque sólo una opción):

Conoce 1 <input type="radio"/>	Conoce más de 1 <input type="radio"/>	Conoce ninguno <input type="radio"/>
--------------------------------	---------------------------------------	--------------------------------------

4. ¿Ha usado alguno de estos símbolos en sus proyectos?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

5. ¿Ha estudiado alguno de estos símbolos en sus lecturas?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

6. Según mi experiencia de trabajar en proyectos con esa notación, reconozco al número ___ que sería la mejor opción para representar <Worker> (rellene con el número 0 para "ninguno").

7. Por su significado, expresividad y lo que me da a entender, me parece que es mejor el número ____ para representar <Worker> (rellene con el número 0 para "ninguno").

8. ¿Ha visto otro símbolo que le parezca mejor para representar <Worker>?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

Si su respuesta fue afirmativa responda la pregunta 9, de lo contrario continúe con la pregunta 10.

9. ¿Cuál es el nombre de la propuesta donde encontró el símbolo?:

10. ¿Le gustaría sugerir un símbolo para representar el clasificador <Worker>?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

Si su respuesta fue afirmativa, envíe su propuesta de símbolo al siguiente correo: mlvillegas@uniquindio.edu.co.

La TxA define el clasificador <Tarea que hace el worker>, en el nivel de Negocio (Nivel independiente de la tecnología). Ejemplo: "El Cajero verifica la información de la cuenta bancaria diligenciada por el Cliente en el formato de "Transacciones"": <Tarea que hace el worker>.

Ahora, conteste las preguntas 1 a 10 de acuerdo a dicha especificación.

1. ¿Ha usado el clasificador <Tarea que hace el worker> en sus proyectos? (marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

2. ¿Ha estudiado el clasificador <Tarea que hace el worker> en sus lecturas? (marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

De acuerdo a la figura C.9 conteste las preguntas 3, 4, 5, 6, 7

3. De los símbolos en la figura C.9 "Tarea que hace el worker", (marque sólo una opción):

Conoce 1 <input type="radio"/>	Conoce más de 1 <input type="radio"/>	Conoce ninguno <input type="radio"/>
--------------------------------	---------------------------------------	--------------------------------------

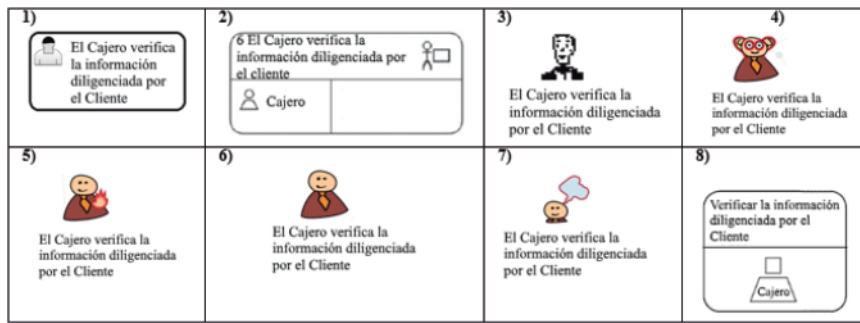


Figura C.9. Símbolos de algunas propuestas existentes para representar - Tarea que hace el Worker

4. ¿Ha usado alguno de estos símbolos en sus proyectos?(*marque sólo una opción*):

Si No

5. ¿Ha estudiado alguno de estos símbolos en sus lecturas?(*marque sólo una opción*):

Si No

6. Según mi experiencia de trabajar en proyectos con esa notación, reconozco al número ___ que sería la mejor opción para representar <Tarea que hace el worker> (*rellene con el número 0 para "ninguno"*).

7. Por su significado, expresividad y lo que me da a entender, me parece que es mejor el número ___ para representar <Tarea que hace el worker> (*rellene con el número 0 para "ninguno"*).

8. ¿Ha visto otro símbolo que le parezca mejor para representar <Tarea que hace el worker>?(*marque sólo una opción*):

Si No

Si su respuesta fue afirmativa responda la pregunta 9, de lo contrario continúe con la pregunta 10.

9. ¿Cuál es el nombre de la propuesta donde encontró el símbolo?:

10. ¿Le gustaría sugerir un símbolo para representar el clasificador <Tarea que hace el worker>?(*marque sólo una opción*):

Si No

La TxA define el clasificador <Tiempo>, en el nivel de Negocio (Nivel independiente de la tecnología). Ejemplo: "Esperar turno": <Tiempo>.

Ahora, conteste las preguntas 1 a 10 de acuerdo a dicha especificación.

1. ¿Ha usado el clasificador <Tiempo> en sus proyectos? (marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

2. ¿Ha estudiado el clasificador <Tiempo> en sus lecturas? (marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

De acuerdo a la figura C.10 conteste las preguntas 3, 4, 5, 6, 7



Figura C.10. Símbolos de algunas propuestas existentes para representar - Tiempo

3. Conoce el símbolo de la figura C.10 "Tiempo", (marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

4. ¿Ha usado el símbolo de la figura C.10 en sus proyectos?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

5. ¿Ha estudiado el símbolo de la figura C.10 en sus lecturas?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

6. Según mi experiencia de trabajar en proyectos con esa notación, reconozco al número ___ que sería la mejor opción para representar <Tiempo> (rellene con el número 0 para "ninguno").

7. Por su significado, expresividad y lo que me da a entender, me parece que es mejor el número ___ para representar <Tiempo> (rellene con el número 0 para "ninguno").

8. ¿Ha visto otro símbolo que le parezca mejor para representar <Tiempo>?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

Si su respuesta fue afirmativa responda la pregunta 9, de lo contrario continúe con la pregunta 10.



9. ¿Cuál es el nombre de la propuesta donde encontró el símbolo?:

10. ¿Le gustaría sugerir un símbolo para representar el clasificador <Tiempo>?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

Si su respuesta fue afirmativa, envíe su propuesta de símbolo al siguiente correo: mlvillegas@uniquindio.edu.co.

La TxA define el clasificador <Awareness de Negocio>, en el nivel de Negocio (Nivel independiente de la tecnología). Ejemplo: "El Cajero está verificando la información": <Awareness de Negocio>.

Ahora, conteste las preguntas 1 a 10 de acuerdo a dicha especificación.

1. ¿Ha usado el clasificador <Awareness de Negocio> en sus proyectos? (marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

2. ¿Ha estudiado el clasificador <Awareness de Negocio> en sus lecturas? (marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

De acuerdo a la figura C.11 conteste las preguntas 3, 4, 5, 6, 7

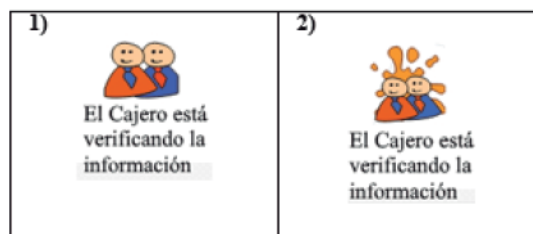


Figura C.11. Símbolos de algunas propuestas existentes para representar - Awareness de negocio

3. De los símbolos en la figura C.11 "Awareness de Negocio", (marque sólo una opción):

Conoce 1 <input type="radio"/>	Conoce más de 1 <input type="radio"/>	Conoce ninguno <input type="radio"/>
--------------------------------	---------------------------------------	--------------------------------------

4. ¿Ha usado alguno de estos símbolos en sus proyectos?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

5. ¿Ha estudiado alguno de estos símbolos en sus lecturas?(*marque sólo una opción*):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

6. Según mi experiencia de trabajar en proyectos con esa notación, reconozco al número ____ que sería la mejor opción para representar <Awareness de Negocio> (*rellene con el número 0 para "ninguno"*).

7. Por su significado, expresividad y lo que me da a entender, me parece que es mejor el número ____ para representar <Awareness de Negocio> (*rellene con el número 0 para "ninguno"*).

8. ¿Ha visto otro símbolo que le parezca mejor para representar <Awareness de Negocio>?(*marque sólo una opción*):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

Si su respuesta fue afirmativa responda la pregunta 9, de lo contrario continúe con la pregunta 10.

9. ¿Cuál es el nombre de la propuesta donde encontró el símbolo?:

10. ¿Le gustaría sugerir un símbolo para representar el clasificador <Awareness de Negocio>?(*marque sólo una opción*):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

Si su respuesta fue afirmativa, envíe su propuesta de símbolo al siguiente correo: mlvillegas@uniquindio.edu.co.

La TxA define el clasificador <Feedback de Negocio>, en el nivel de Negocio (Nivel independiente de la tecnología). Ejemplo: "El Cajero informa al cliente que el formulario está correctamente diligenciado": <Feedback de Negocio>.

Ahora, conteste las preguntas 1 a 10 de acuerdo a dicha especificación.

1. ¿Ha usado el clasificador <Feedback de Negocio> en sus proyectos? (*marque sólo una opción*):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

2. ¿Ha estudiado el clasificador <Feedback de Negocio> en sus lecturas? (*marque sólo una opción*):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

De acuerdo a la figura C.12 conteste las preguntas 3, 4, 5, 6, 7

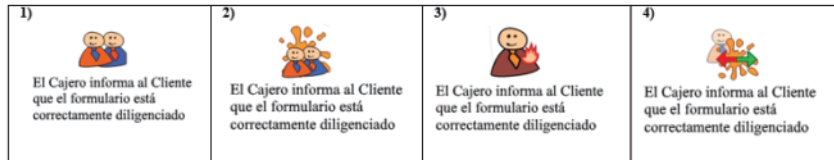


Figura C.12. Símbolos de algunas propuestas existentes para representar - Feedback de Negocio

3. De los símbolos en la figura C.12 "Feedback de Negocio", (marque sólo una opción):

Conoce 1 <input type="radio"/>	Conoce más de 1 <input type="radio"/>	Conoce ninguno <input type="radio"/>
--------------------------------	---------------------------------------	--------------------------------------

4. ¿Ha usado alguno de estos símbolos en sus proyectos?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

5. ¿Ha estudiado alguno de estos símbolos en sus lecturas?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

6. Según mi experiencia de trabajar en proyectos con esa notación, reconozco al número ____ que sería la mejor opción para representar <Feedback de Negocio> (rellene con el número 0 para "ninguno").

7. Por su significado, expresividad y lo que me da a entender, me parece que es mejor el número ____ para representar <Feedback de Negocio> (rellene con el número 0 para "ninguno").

8. ¿Ha visto otro símbolo que le parezca mejor para representar <Feedback de Negocio>?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

Si su respuesta fue afirmativa responda la pregunta 9, de lo contrario continúe con la pregunta 10.

9. ¿Cuál es el nombre de la propuesta donde encontró el símbolo?:

10. ¿Le gustaría sugerir un símbolo para representar el clasificador <Feedback de Negocio>?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

Si su respuesta fue afirmativa, envíe su propuesta de símbolo al siguiente correo: mlvillegas@uniquindio.edu.co.



SELECCIÓN DE ELEMENTOS DE MODELADO DE SISTEMAS INTERACTIVOS

C.2 NIVEL DE SISTEMA

Con esta encuesta se pretende seleccionar un conjunto de elementos notacionales para representar los clasificadores definidos en la Taxonomía de la Actividad para Sistemas Interactivos.

Dinámica para contestar la encuesta

La dinámica que se seguirá con esta encuesta es: Se presenta cada clasificador definido en la Taxonomía de la Actividad (TxA) y un fragmento de la especificación para la Actividad "Retirar Dinero" que relacione dicho clasificador. El encuestado analiza y contesta las preguntas de acuerdo a lo que se plantea en el fragmento de la especificación. Se debe consultar simultáneamente el documento del Contexto.

1. Nombre del encuestado: _____

La TxA define el clasificador <Tarea que hace el Usuario>, en el nivel de Sistema.
Ejemplo: "El Cajero decide qué tipo de transacción va a realizar en el sistema": <Tarea que hace el Usuario>.

Ahora, conteste las preguntas 1 a 10 de acuerdo a dicha especificación.

1. ¿Ha usado el clasificador <Tarea que hace el Usuario> en sus proyectos? (*marque sólo una opción*):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

2. ¿Ha estudiado el clasificador <Tarea que hace el Usuario> en sus lecturas? (*marque sólo una opción*):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

De acuerdo a la figura C.13 conteste las preguntas 3, 4, 5, 6, 7

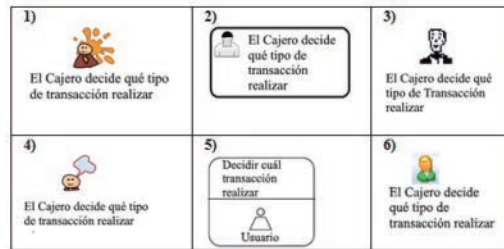


Figura C.13. Símbolos de algunas propuestas existentes para representar - Tarea que hace el Usuario.

3. De los símbolos en la figura C.13 "*Tarea que hace el Usuario*", (*marque sólo una opción*):

Conoce 1 <input type="radio"/>	Conoce más de 1 <input type="radio"/>	Conoce ninguno <input type="radio"/>
--------------------------------	---------------------------------------	--------------------------------------

4. ¿Ha usado alguno de estos símbolos en sus proyectos?(*marque sólo una opción*):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

5. ¿Ha estudiado alguno de estos símbolos en sus lecturas?(*marque sólo una opción*):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

6. Según mi experiencia de trabajar en proyectos con esa notación, reconozco al número ____ que sería la mejor opción para representar <*Tarea que hace el Usuario*> (*rellene con el número 0 para "ninguno"*).

7. Por su significado, expresividad y lo que me da a entender, me parece que es mejor el número ____ para representar <*Tarea que hace el Usuario*> (*rellene con el número 0 para "ninguno"*).

8. ¿Ha visto otro símbolo que le parezca mejor para representar <*Tarea que hace el Usuario*>?(*marque sólo una opción*):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

Si su respuesta fue afirmativa responda la pregunta 9, de lo contrario continúe con la pregunta 10.

9. ¿Cuál es el nombre de la propuesta donde encontró el símbolo?:

10. ¿Le gustaría sugerir un símbolo para representar el clasificador <*Tarea que hace el Usuario*>?(*marque sólo una opción*):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

Si su respuesta fue afirmativa, envíe su propuesta de símbolo al siguiente correo: mlvillegas@uniquindio.edu.co.

La TxA define el clasificador <Usuario>, en el nivel de Sistema. Ejemplo: "Cajero": <Usuario>.

Ahora, conteste las preguntas 1 a 10 de acuerdo a dicha especificación.

1. ¿Ha usado el clasificador <Usuario> en sus proyectos? (marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

2. ¿Ha estudiado el clasificador <Usuario> en sus lecturas? (marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

De acuerdo a la figura C.14 conteste las preguntas 3, 4, 5, 6, 7

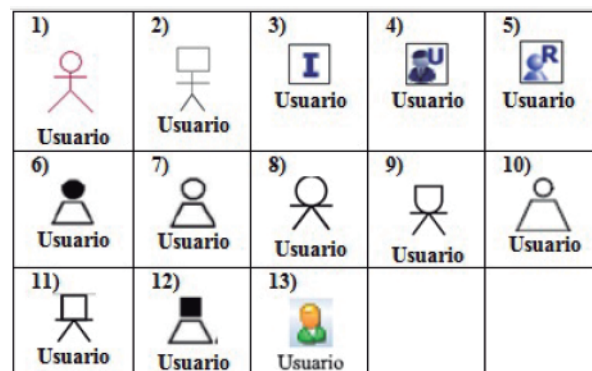


Figura C.14. Símbolos de algunas propuestas existentes para representar - Usuario.

3. De los símbolos en la figura C.14 "Usuario", (marque sólo una opción):

Conoce 1 <input type="radio"/>	Conoce más de 1 <input type="radio"/>	Conoce ninguno <input type="radio"/>
--------------------------------	---------------------------------------	--------------------------------------

4. ¿Ha usado alguno de estos símbolos en sus proyectos?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

5. ¿Ha estudiado alguno de estos símbolos en sus lecturas?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

6. Según mi experiencia de trabajar en proyectos con esa notación, reconozco al



número ____ que sería la mejor opción para representar <Usuario> (rellene con el número 0 para "ninguno").

7. Por su significado, expresividad y lo que me da a entender, me parece que es mejor el número ____ para representar <Usuario> (rellene con el número 0 para "ninguno").

8. ¿Ha visto otro símbolo que le parezca mejor para representar <Usuario>?(marque sólo una opción):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

Si su respuesta fue afirmativa responda la pregunta 9, de lo contrario continúe con la pregunta 10.

9. ¿Cuál es el nombre de la propuesta donde encontró el símbolo?:

10. ¿Le gustaría sugerir un símbolo para representar el clasificador <Usuario>?(marque sólo una opción):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

Si su respuesta fue afirmativa, envíe su propuesta de símbolo al siguiente correo: mlvillegas@uniquindio.edu.co.

La TxA define el clasificador <Ubicación>, en el nivel de Sistema. Ejemplo: "Sistema Central": <Ubicación de Sistema>.

Ahora, conteste las preguntas 1 a 10 de acuerdo a dicha especificación.

1. ¿Ha usado el clasificador <Ubicación> a nivel de sistema en sus proyectos? (marque sólo una opción):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

2. ¿Ha estudiado el clasificador <Ubicación> a nivel de sistema en sus lecturas? (marque sólo una opción):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

De acuerdo a la figura C.15 conteste las preguntas 3, 4, 5, 6, 7

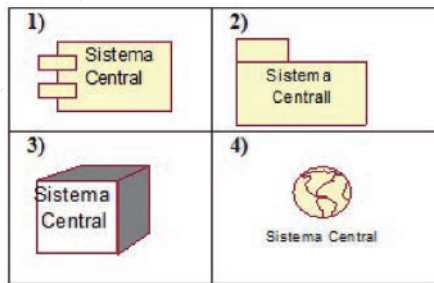


Figura C.15. Símbolos de algunas propuestas existentes para representar - Ubicación de Sistema.

3. De los símbolos en la figura C.15 "Ubicación de Sistema", (marque sólo una opción):

Conoce 1 <input type="radio"/>	Conoce más de 1 <input type="radio"/>	Conoce ninguno <input type="radio"/>
--------------------------------	---------------------------------------	--------------------------------------

4. ¿Ha usado alguno de estos símbolos en sus proyectos?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

5. ¿Ha estudiado alguno de estos símbolos en sus lecturas?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

6. Según mi experiencia de trabajar en proyectos con esa notación, reconozco al número ___ que sería la mejor opción para representar <Ubicación de sistema> (rellene con el número 0 para "ninguno").

7. Por su significado, expresividad y lo que me da a entender, me parece que es mejor el número ___ para representar <Ubicación de Sistema> (rellene con el número 0 para "ninguno").

8. ¿Ha visto otro símbolo que le parezca mejor para representar <Ubicación de Sistema>?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

Si su respuesta fue afirmativa responda la pregunta 9, de lo contrario continúe con la pregunta 10.

9. ¿Cuál es el nombre de la propuesta donde encontró el símbolo?:

10. ¿Le gustaría sugerir un símbolo para representar el clasificador <Ubicación de Sistema>?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------



Si su respuesta fue afirmativa, envíe su propuesta de símbolo al siguiente correo: mlvillegas@uniquindio.edu.co.

La TxA define el clasificador <Dato>, en el nivel de Sistema. Ejemplo: "Transacción": <Dato de Sistema>.

Ahora, conteste las preguntas 1 a 10 de acuerdo a dicha especificación.

1. ¿Ha usado el clasificador <Dato de Sistema> en sus proyectos? (marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

2. ¿Ha estudiado el clasificador <Dato de Sistema> en sus lecturas? (marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

De acuerdo a la figura C.16 conteste las preguntas 3, 4, 5, 6, 7

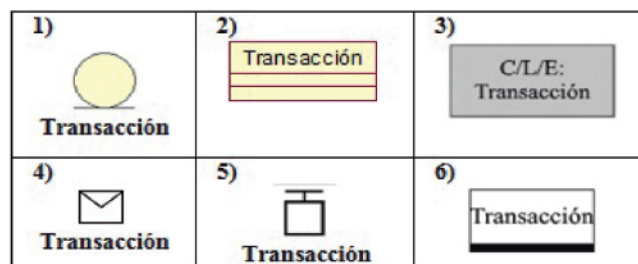


Figura C.16. Símbolos de algunas propuestas existentes para representar - Dato de Sistema.

3. De los símbolos en la figura C.16 "Dato de Sistema", (marque sólo una opción):

Conoce 1 <input type="radio"/>	Conoce más de 1 <input type="radio"/>	Conoce ninguno <input type="radio"/>
--------------------------------	---------------------------------------	--------------------------------------

4. ¿Ha usado alguno de estos símbolos en sus proyectos?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

5. ¿Ha estudiado alguno de estos símbolos en sus lecturas?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

6. Según mi experiencia de trabajar en proyectos con esa notación, reconozco al número ____ que sería la mejor opción para representar <Dato de Sistema> (rellene con el número 0 para "ninguno").

7. Por su significado, expresividad y lo que me da a entender, me parece que es mejor el número ____ para representar <Dato de Sistema> (rellene con el número 0 para "ninguno").

8. ¿Ha visto otro símbolo que le parezca mejor para representar <Dato de Sistema>?(marque sólo una opción):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

Si su respuesta fue afirmativa responda la pregunta 9, de lo contrario continúe con la pregunta 10.

9. ¿Cuál es el nombre de la propuesta donde encontró el símbolo?:

10. ¿Le gustaría sugerir un símbolo para representar el clasificador <Dato de Sistema>?(marque sólo una opción):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

Si su respuesta fue afirmativa, envíe su propuesta de símbolo al siguiente correo: mlvillegas@uniquindio.edu.co.

La TxA define el clasificador <Interacción>, en el nivel de Sistema. Ejemplo: "El Cajero selecciona la opción "Retiro de Dinero"": <Interacción>.

Ahora, conteste las preguntas 1 a 10 de acuerdo a dicha especificación.

1. ¿Ha usado el clasificador <Interacción> en sus proyectos? (marque sólo una opción):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

2. ¿Ha estudiado el clasificador <Interacción> en sus lecturas? (marque sólo una opción):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

De acuerdo a la figura C.17 conteste las preguntas 3, 4, 5, 6, 7

3. De los símbolos en la figura C.17 "Interacción", (marque sólo una opción):

Conoce 1	<input type="radio"/>	Conoce más de 1	<input type="radio"/>	Conoce ninguno	<input type="radio"/>
----------	-----------------------	-----------------	-----------------------	----------------	-----------------------

4. ¿Ha usado alguno de estos símbolos en sus proyectos?(marque sólo una opción):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

5. ¿Ha estudiado alguno de estos símbolos en sus lecturas?(marque sólo una opción):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

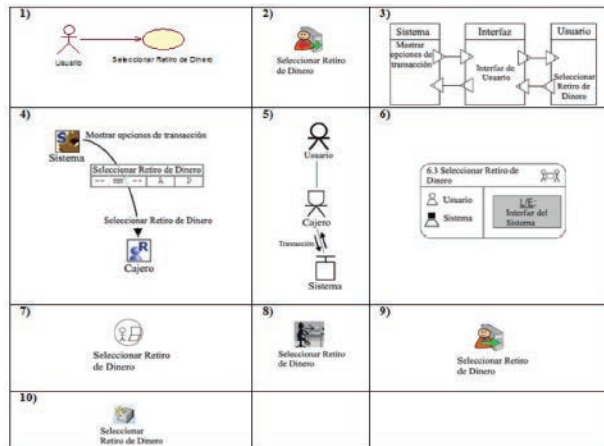


Figura C.17. Símbolos de algunas propuestas existentes para representar - Interacción

6. Según mi experiencia de trabajar en proyectos con esa notación, reconozco al número ____ que sería la mejor opción para representar <Interacción> (rellene con el número 0 para “ninguno”).

7. Por su significado, expresividad y lo que me da a entender, me parece que es mejor el número ____ para representar <Interacción> (rellene con el número 0 para “ninguno”).

8. ¿Ha visto otro símbolo que le parezca mejor para representar <Interacción>?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

Si su respuesta fue afirmativa responda la pregunta 9, de lo contrario continúe con la pregunta 10.

9. ¿Cuál es el nombre de la propuesta donde encontró el símbolo?:

10. ¿Le gustaría sugerir un símbolo para representar el clasificador <Interacción>?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

Si su respuesta fue afirmativa, envíe su propuesta de símbolo al siguiente correo: mlvillegas@uniquindio.edu.co.

La TxA define el clasificador <Objetivo de Sistema>, en el nivel de Sistema. Ejemplo: "La Transacción queda comprometida en el Sistema Local y en el Sistema Central": <Objetivo de Sistema>.

Ahora, conteste las preguntas 1 a 10 de acuerdo a dicha especificación.

1. ¿Ha usado el clasificador <Objetivo de Sistema> en sus proyectos? (marque sólo una opción):

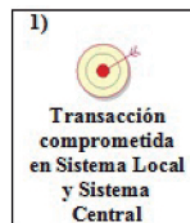
Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

2. ¿Ha estudiado el clasificador <Objetivo de Sistema> en sus lecturas? (marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

De acuerdo a la figura C.18 conteste las preguntas 3, 4, 5, 6, 7

Figura C.18. Símbolos de algunas propuestas existentes para representar - Objetivo de Sistema.



3. Conoce el símbolo de la figura C.18 "Objetivo de Sistema", (marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

4. ¿Ha usado este símbolo en sus proyectos?(marque sólo una opción):

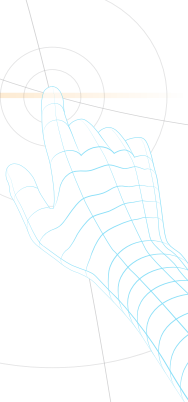
Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

5. ¿Ha estudiado este símbolo en sus lecturas?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

6. Según mi experiencia de trabajar en proyectos con esa notación, reconozco al número ____ que sería la mejor opción para representar <Objetivo de Sistema> (rellene con el número 0 para "ninguno").

7. Por su significado, expresividad y lo que me da a entender, me parece que es mejor el número ____ para representar <Objetivo de Sistema> (rellene con el número 0 para "ninguno").



8. ¿Ha visto otro símbolo que le parezca mejor para representar <Objetivo de Sistema>?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

Si su respuesta fue afirmativa responda la pregunta 9, de lo contrario continúe con la pregunta 10.

9. ¿Cuál es el nombre de la propuesta donde encontró el símbolo?:

10. ¿Le gustaría sugerir un símbolo para representar el clasificador <Objetivo de Sistema>?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

Si su respuesta fue afirmativa, envíe su propuesta de símbolo al siguiente correo: mlvillegas@uniquindio.edu.co.

La TxA define el clasificador <Regla de Sistema>, en el nivel de Sistema. Ejemplo: "Si el monto es inferior a la suma permitida para retiro a través de cajero automático, se cobra una comisión adicional al Cliente": <Regla de Sistema>.

Ahora, conteste las preguntas 1 a 10 de acuerdo a dicha especificación.

1. ¿Ha usado el clasificador <Regla de Sistema> en sus proyectos? (marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

2. ¿Ha estudiado el clasificador <Regla de Sistema> en sus lecturas? (marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

De acuerdo a la figura C.19 conteste las preguntas 3, 4, 5, 6, 7

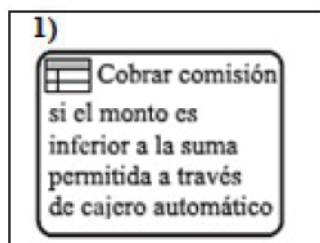


Figura C.19. Símbolos de algunas propuestas existentes para representar - Regla de Sistema.

3. Conoce el símbolo de la figura C.19 "Regla de sistema", (marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

4. ¿Ha usado este símbolo en sus proyectos?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

5. ¿Ha estudiado este símbolo en sus lecturas?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

6. Según mi experiencia de trabajar en proyectos con esa notación, reconozco al número ____ que sería la mejor opción para representar <Regla de sistema> (rellene con el número 0 para "ninguno").

7. Por su significado, expresividad y lo que me da a entender, me parece que es mejor el número ____ para representar <Regla de Sistema> (rellene con el número 0 para "ninguno").

8. ¿Ha visto otro símbolo que le parezca mejor para representar <Regla de Sistema>?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

Si su respuesta fue afirmativa responda la pregunta 9, de lo contrario continúe con la pregunta 10.

9. ¿Cuál es el nombre de la propuesta donde encontró el símbolo?:

10. ¿Le gustaría sugerir un símbolo para representar el clasificador <Regla de Sistema>?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

Si su respuesta fue afirmativa, envíe su propuesta de símbolo al siguiente correo: mlvillegas@uniquindio.edu.co.

La TxA define el clasificador <Sistema>, en el nivel de Sistema. Ejemplo: "Sistema": <Sistema>.

Ahora, conteste las preguntas 1 a 10 de acuerdo a dicha especificación.

1. ¿Ha usado el clasificador <Sistema> en sus proyectos? (marque sólo una opción):

2. ¿Ha estudiado el clasificador <Sistema> en sus lecturas? (marque sólo una opción):

De acuerdo a la figura C.20 conteste las preguntas 3, 4, 5, 6, 7

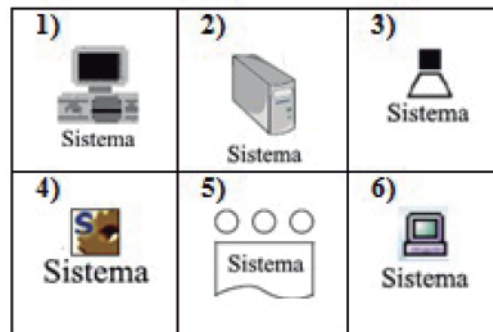


Figura C.19. Símbolos de algunas propuestas existentes para representar - Sistema

3. De los símbolos en la figura C.20 "Sistema", (marque sólo una opción):

Conoce 1 <input type="radio"/>	Conoce más de 1 <input type="radio"/>	Conoce ninguno <input type="radio"/>
--------------------------------	---------------------------------------	--------------------------------------

4. ¿Ha usado alguno de estos símbolos en sus proyectos?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

5. ¿Ha estudiado alguno de estos símbolos en sus lecturas?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

6. Según mi experiencia de trabajar en proyectos con esa notación, reconozco al número ___ que sería la mejor opción para representar <Sistema> (rellene con el número 0 para "ninguno").

7. Por su significado, expresividad y lo que me da a entender, me parece que es mejor el número ____ para representar <Sistema> (rellene con el número 0 para "ninguno").

8. ¿Ha visto otro símbolo que le parezca mejor para representar <Sistema>?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

Si su respuesta fue afirmativa responda la pregunta 9, de lo contrario continúe con la pregunta 10.

9. ¿Cuál es el nombre de la propuesta donde encontró el símbolo?:

10. ¿Le gustaría sugerir un símbolo para representar el clasificador <Sistema>?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

Si su respuesta fue afirmativa, envíe su propuesta de símbolo al siguiente correo: mlvillegas@uniquindio.edu.co.

La TxA define el clasificador <Tarea que hace el Sistema>, en el nivel de Sistema.
Ejemplo: "El Sistema valida la información (Si existe la cuenta, si existe el cliente)":
<Tarea que hace el Sistema>.

Ahora, conteste las preguntas 1 a 10 de acuerdo a dicha especificación.

1. ¿Ha usado el clasificador <Tarea que hace el Sistema> en sus proyectos? (marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

2. ¿Ha estudiado el clasificador <Tarea que hace el Sistema> en sus lecturas? (marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

De acuerdo a la figura C.21 conteste las preguntas 3, 4, 5, 6, 7

3. De los símbolos en la figura C.21 "Tarea que hace el Sistema", (marque sólo una opción):

Conoce 1 <input type="radio"/>	Conoce más de 1 <input type="radio"/>	Conoce ninguno <input type="radio"/>
--------------------------------	---------------------------------------	--------------------------------------

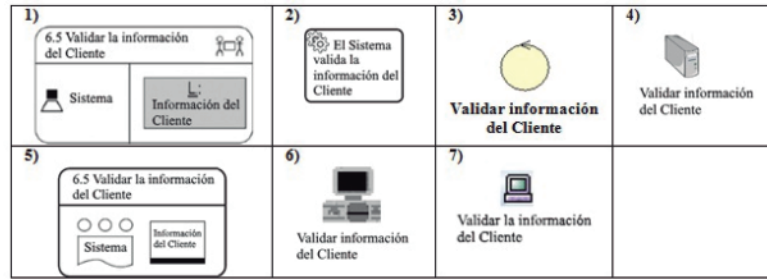


Figura C.21. Símbolos de algunas propuestas existentes para representar - Tarea que hace el Sistema

4. ¿Ha usado alguno de estos símbolos en sus proyectos?(*marque sólo una opción*):

Si No

5. ¿Ha estudiado alguno de estos símbolos en sus lecturas?(*marque sólo una opción*):

Si No

6. Según mi experiencia de trabajar en proyectos con esa notación, reconozco al número ___ que sería la mejor opción para representar <Tarea que hace el Sistema> (*rellene con el número 0 para "ninguno"*).

7. Por su significado, expresividad y lo que me da a entender, me parece que es mejor el número ___ para representar <Tarea que hace el Sistema> (*rellene con el número 0 para "ninguno"*).

8. ¿Ha visto otro símbolo que le parezca mejor para representar <Tarea que hace el Sistema>?(*marque sólo una opción*):

Si No

Si su respuesta fue afirmativa responda la pregunta 9, de lo contrario continúe con la pregunta 10.

9. ¿Cuál es el nombre de la propuesta donde encontró el símbolo?:

10. ¿Le gustaría sugerir un símbolo para representar el clasificador <Tarea que hace el Sistema>?(*marque sólo una opción*):

Si No

La TxA define el clasificador <Tiempo>, en el nivel de Sistema. Ejemplo: "El Sistema tarda un tiempo en realizar la transacción": <Tiempo>.

Ahora, conteste las preguntas 1 a 10 de acuerdo a dicha especificación.

1. ¿Ha usado el clasificador <Tiempo> a nivel de sistema en sus proyectos? (*marque sólo una opción*):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

2. ¿Ha estudiado el clasificador <Tiempo> a nivel de sistema en sus lecturas? (*marque sólo una opción*):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

De acuerdo a la figura C.22 conteste las preguntas 3, 4, 5, 6, 7



Figura C.22. Símbolos de algunas propuestas existentes para representar - Tiempo

3. Conoce el símbolo de la figura C.22 "Tiempo", (*marque sólo una opción*):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

4. ¿Ha usado el símbolo de la figura C.22 en sus proyectos? (*marque sólo una opción*):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

5. ¿Ha estudiado el símbolo de la figura C.22 en sus lecturas? (*marque sólo una opción*):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

6. Según mi experiencia de trabajar en proyectos con esa notación, reconozco al número ____ que sería la mejor opción para representar <Tiempo> (*rellene con el número 0 para "ninguno"*).

7. Por su significado, expresividad y lo que me da a entender, me parece que es mejor el número ____ para representar <Tiempo> (*rellene con el número 0 para "ninguno"*).

8. ¿Ha visto otro símbolo que le parezca mejor para representar <Tiempo>? (*marque sólo una opción*):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

Si su respuesta fue afirmativa responda la pregunta 9, de lo contrario continúe con la pregunta 10.



9. ¿Cuál es el nombre de la propuesta donde encontró el símbolo?:

10. ¿Le gustaría sugerir un símbolo para representar el clasificador <Tiempo>?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

Si su respuesta fue afirmativa, envíe su propuesta de símbolo al siguiente correo: mlvillegas@uniquindio.edu.co.

La TxA define el clasificador <Awareness de Sistema>, en el nivel de Sistema.
Ejemplo: "El Sistema debe informar al Usuario (Cajero) que está realizando la transacción": <Awareness de sistema>.

Ahora, conteste las preguntas 1 a 10 de acuerdo a dicha especificación.

1. ¿Ha usado el clasificador <Awareness de sistema> en sus proyectos? (marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

2. ¿Ha estudiado el clasificador <Awareness de sistema> en sus lecturas? (marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

De acuerdo a la figura C.23 conteste las preguntas 3, 4, 5, 6, 7

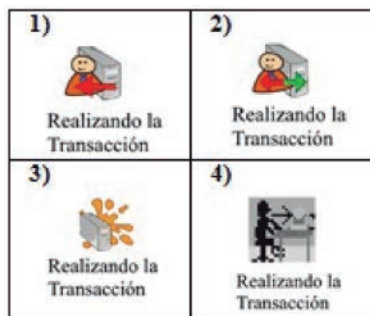


Figura C.23. Símbolos de algunas propuestas existentes para representar - Awareness de sistema

3. De los símbolos en la figura C.23 "Awareness de Sistema", (marque sólo una opción):

Conoce 1 <input type="radio"/>	Conoce más de 1 <input type="radio"/>	Conoce ninguno <input type="radio"/>
--------------------------------	---------------------------------------	--------------------------------------

4. ¿Ha usado alguno de estos símbolos en sus proyectos?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

5. ¿Ha estudiado alguno de estos símbolos en sus lecturas?(*marque sólo una opción*):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

6. Según mi experiencia de trabajar en proyectos con esa notación, reconozco al número ____ que sería la mejor opción para representar <Awareness de sistema> (*rellene con el número 0 para "ninguno"*).

7. Por su significado, expresividad y lo que me da a entender, me parece que es mejor el número ____ para representar <Awareness de sistema> (*rellene con el número 0 para "ninguno"*).

8. ¿Ha visto otro símbolo que le parezca mejor para representar <Awareness de sistema>?(*marque sólo una opción*):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

Si su respuesta fue afirmativa responda la pregunta 9, de lo contrario continúe con la pregunta 10.

9. ¿Cuál es el nombre de la propuesta donde encontró el símbolo?:

10. ¿Le gustaría sugerir un símbolo para representar el clasificador <Awareness de Negocio>?(*marque sólo una opción*):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

Si su respuesta fue afirmativa, envíe su propuesta de símbolo al siguiente correo: mlvillegas@uniquindio.edu.co.

La TxA define el clasificador <Feedback de sistema>, en el nivel de Negocio (Nivel independiente de la tecnología). Ejemplo: "El Sistema debe informar al Usuario (Cajero) que comprometió la transacción": <Feedback de Sistema>.

Ahora, conteste las preguntas 1 a 10 de acuerdo a dicha especificación.

1. ¿Ha usado el clasificador <Feedback de Sistema> en sus proyectos? (*marque sólo una opción*):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

2. ¿Ha estudiado el clasificador <Feedback de Sistema> en sus lecturas? (*marque sólo una opción*):

Si	<input type="radio"/>	No	<input type="radio"/>
----	-----------------------	----	-----------------------

De acuerdo a la figura C.24 conteste las preguntas 3, 4, 5, 6, 7

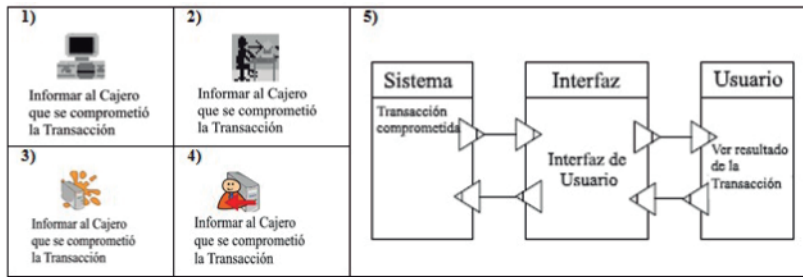


Figura C.24. Símbolos de algunas propuestas existentes para representar - Feedback de Sistema

3. De los símbolos en la figura C.24 "Feedback de Sistema", (marque sólo una opción):

Conoce 1 <input type="radio"/>	Conoce más de 1 <input type="radio"/>	Conoce ninguno <input type="radio"/>
--------------------------------	---------------------------------------	--------------------------------------

4. ¿Ha usado alguno de estos símbolos en sus proyectos?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

5. ¿Ha estudiado alguno de estos símbolos en sus lecturas?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

6. Según mi experiencia de trabajar en proyectos con esa notación, reconozco al número ____ que sería la mejor opción para representar <Feedback de Sistema> (rellene con el número 0 para "ninguno").

7. Por su significado, expresividad y lo que me da a entender, me parece que es mejor el número ____ para representar <Feedback de Sistema> (rellene con el número 0 para "ninguno").

8. ¿Ha visto otro símbolo que le parezca mejor para representar <Feedback de Sistema>?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

Si su respuesta fue afirmativa responda la pregunta 9, de lo contrario continúe con la pregunta 10.

9. ¿Cuál es el nombre de la propuesta donde encontró el símbolo?:

10. ¿Le gustaría sugerir un símbolo para representar el clasificador <Feedback de Sistema>?(marque sólo una opción):

Si <input type="radio"/>	No <input type="radio"/>
--------------------------	--------------------------

Si su respuesta fue afirmativa, envíe su propuesta de símbolo al siguiente correo: mlvillegas@uniquindio.edu.co.



ANEXO D

NOTACIÓN QUE SOPORTA EL MODELADO DE LA ACTIVIDAD PARA MBUID

En esta sección se presentan los elementos de modelado que han sido definidos para soportar el diseño de la interfaz de usuario para sistemas interactivos, basado en el modelado de la actividad. Como ya se ha mencionado, la notación definida parte de las propuestas: Marco de Desarrollo CIAF (Giraldo, 2010) y CIAN (Molina, 2008).

D.1 MODELADO A NIVEL DE NEGOCIO

Este nivel comprende los siguientes diagramas: Diagrama de Plantilla de Proceso (*Process Template*), Diagrama de Interfaz de Usuario de Negocio (*Business User Interface Diagram*) y Diagrama de Entidades de Negocio (*Business Entity Diagram*). (Tabla D.1)

Diagrama de Plantilla de Proceso de Negocio (*Process Template*)

Contiene la especificación de la Interacción entre los actores del negocio (*Business Interaction*) y la especificación de las tareas interactivas de negocio (*Business Interactive Task*).

Diagrama de Interacción de Negocio (*Business Interaction*)

Describe la interacción entre el Cliente y el (los) Worker (s) a través de formularios en papel (*interaction*) y también la interacción con las entidades del negocio (*access*). La Tabla D.2 describe los elementos de modelado para este diagrama.

Diagrama de Tareas Interactivas de Negocio (*Business Interactive Task*)

Describe la secuencia de tareas interactivas de negocio. Se soportan dos niveles de granularidad para dichas tareas interactivas, es decir, es posible especificar Tareas que se componen de Tareas Individuales, Tareas Cooperativas y/o Tareas de Comunicación. La Tabla D.3 describe los elementos de modelado para este diagrama.

D.2 MODELADO A NIVEL DE SISTEMA

Este nivel comprende los siguientes diagramas: Patrón de Presentación (*Presentation Pattern*) y Diagrama de Navegación (*Navigation Diagram*). (Tabla D.4)

Tabla D.1. Elementos de modelado para MBUID, nivel de negocio

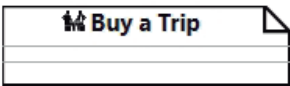
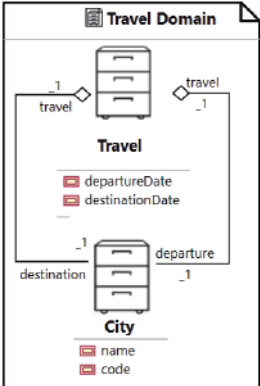
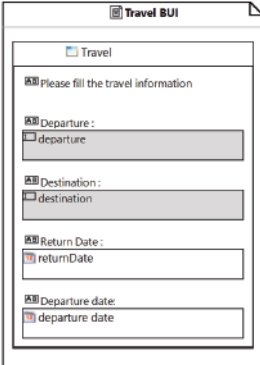

Elementos de modelado nivel de negocio	
	<p>Process Template: Se compone de tres compartimientos. El primero contiene el nombre del proceso. El segundo contiene el Diagrama de Interacción de Negocio (<i>Business Interaction Diagram</i>) y el tercero contiene el Diagrama de Tareas de Negocio (<i>Business Interactive Task Diagram</i>).</p>
	<p>Business Entity Diagram (BED): Se compone de dos compartimientos. El primero contiene el nombre del diagrama y el segundo contiene el diagrama de Entidades de Negocio.</p>
	<p>Business User Interface Diagram (BUID): Se compone de dos compartimientos. El primero contiene el nombre del diagrama y el segundo contiene el diagrama de Interfaz de Usuario de Negocio.</p>
	<p>Mapping Relationship: Es la relación que existe entre el Process Template y los Diagramas BUID y BED.</p>

Diagrama de Patrón de Presentación (Presentation Pattern)

Contiene el conjunto de diagramas necesarios para especificar, a nivel del sistema interactivo, cada una de las tareas simples especificadas en la Plantilla de Proceso, a nivel de negocio. Cada una de estas tareas simples tiene una trazabilidad hacia los siguientes diagramas: 1. Diagrama de Tareas Interactivas (*Interactive Task Diagram*), 2. Diagrama de Diálogo (*Dialog Diagram*), 3. Vista de la Interfaz Concreta (*View Part Diagram*), 4. Vista del Modelo de Datos (*View Model Diagram*), 5. Modelo de Dominio (*Domain Model Diagram*), 6. Diagramas de Estados (*StateChart Diagram*). Para cada una de las tareas simples especificadas en la Plantilla de Proceso, es preciso crear un Patrón de Presentación.

Tabla D.2. Elementos de modelado para Business Interaction









Elementos de modelado BUID	
	BUID: Describe la interacción entre el Cliente y el Worker.
 Cliente	<i>Cliente</i>
 Travel Agent	<i>Worker</i>
 Payment	<i>Formulario:</i> Medio a través del cual se lleva a cabo la interacción a nivel de negocio.
 Offer	<i>Repositorio de datos</i>
	<i>Comunicación</i>
	<i>Acceso</i>
	<i>Relación</i>

Diagrama de Navegación (Navigation Diagram)

El diagrama de Navegación presenta la interacción existente entre los patrones de presentación, mediante las acciones que permiten conmutar el sistema interactivo desde una tarea hacia las demás tareas interactivas. Sirve como un elemento de orquestación entre la especificación de los modelos mencionados.

Diagrama de Tareas Interactivas (Interactive Task Diagram)

Se conserva la estructura y relaciones temporales de la notación CTT. Este diagrama indica explícitamente el usuario que realiza las tareas. La Tabla D.5 describe los elementos de modelado para este diagrama.

Tabla D.3. Elementos de modelado para Business Interactive Task.

Elementos de modelado Business Interactive Task	
	<p><i>Business Interactive Task:</i> Nivel bajo de granularidad. Las tareas en este nivel pueden ser de tipo “Composite” o “Simple”. Cada diagrama inicia con una tarea de tipo “Inicio” y finaliza con una tarea de tipo “Fin”. Se utilizan las mismas relaciones temporales definidas por la notación CTT. Para el nivel alto de granularidad, el primer compartimiento contiene el nombre de la tarea, el segundo compartimiento contiene un BUID, y el tercer compartimiento contiene un diagrama de tareas conformado por tareas de tipo simple.</p>
	<p><i>Business Interactive Task:</i> Nivel alto de granularidad. Las tareas en este nivel pueden ser sólo de tipo “Simple” (Simple Individual o Comunicación).</p>
	<p><i>Tarea de Comunicación:</i> Especifica comunicación entre personas. Se compone de los campos “nombre de la tarea”, “Roles que se comunican”, “Mapping hacia el BUID”, “Mapping hacia el BED” y “Texto que describe los pasos para llevar a cabo la tarea”.</p>
	<p><i>Mapping Relationship:</i> Denota la relación entre la tarea de comunicación y sus BUID y BED.</p>

Diagrama de Diálogo (Dialog Diagram)

Conserva la estructura del Diagrama de Tareas, especificando además para cada tarea, el tipo de Componentes de Interacción Abstractos (AIC) asociados a la tarea. a Tabla D.6 describe los elementos de modelado para este diagrama.

Model-View -View -Model Pattern

la Vista representa la estructura, despliegue y la apariencia de la Interfaz de Usuario Concreta con sus componentes. El Modelo es una representación de los datos que soportan la especificación de la Interfaz Concreta. Tiene trazabilidad con el Diagrama de Entidades de Negocio. Por último, la Vista del Modelo representan los datos que tienen un mapeo directo hacia la Interfaz Concreta o Vista. La Tabla D.7 describe los elementos de modelado para este diagrama.

Tabla D.4. Elementos de modelado de MBUID a nivel de Sistema.

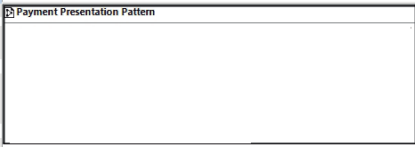
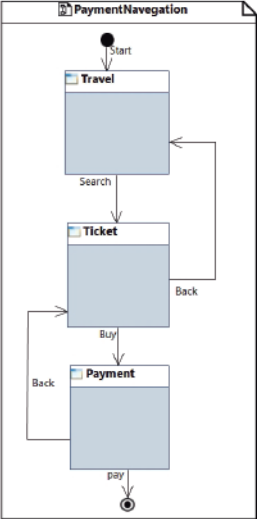
Elementos de modelado Nivel de Sistema	
	<p><i>Presentation Pattern:</i></p> <p>Se compone de dos compartimientos. El primero contiene el nombre del patrón, que corresponde con el nombre de la tarea de negocio a modelar. El segundo contiene el conjunto de diagramas que se describieron anteriormente.</p>
	<p><i>Navigation Diagram:</i> Al igual que en el diagrama de Tareas Interactivas de Negocio, cada diagrama inicia con una tarea de tipo "Inicio" y finaliza con una tarea de tipo "Fin". Cada nodo del diagrama corresponde con un patrón de presentación. Las transiciones entre nodos están dadas por acciones de la interfaz de usuario.</p>

Tabla D.5. Elementos de modelado para Diagrama de Tareas Interactivas a nivel de Sistema.

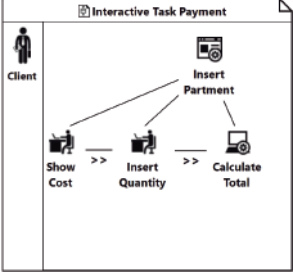



Elementos de modelado Interactive Task Diagram	
	<p><i>Interactive Task Diagram:</i></p> <p>Contiene la estructura jerárquica de las tareas interactivas y también denota qué rol es el responsable de ejecutar las tareas.</p>
 Insert Partment	<p><i>Tarea abstracta</i></p>
 Show Cost	<p><i>Tarea Interactiva</i></p>
 Calculate Total	<p><i>Tarea de Sistema</i></p>

Tabla D.6. Elementos de modelado para Dialog Diagram

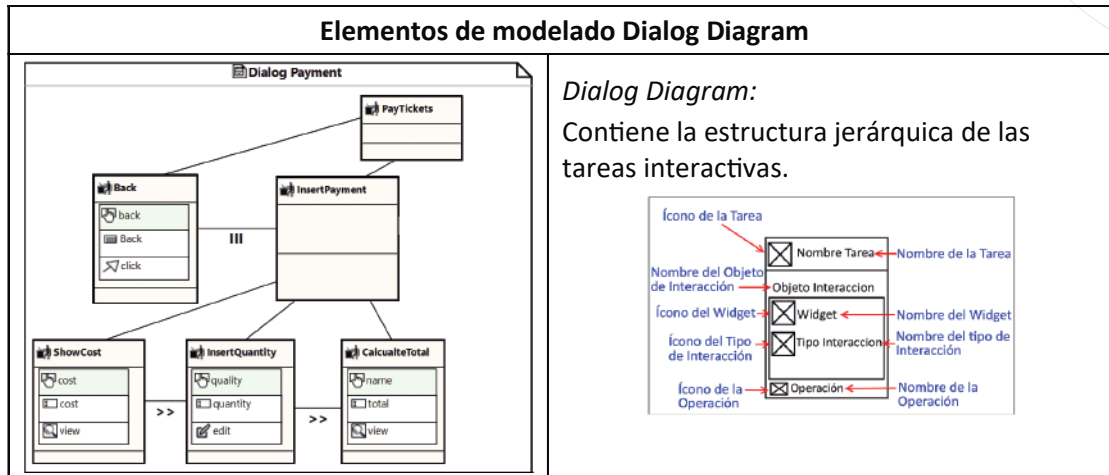


Tabla D.7. Elementos de modelado para View Part-View Model

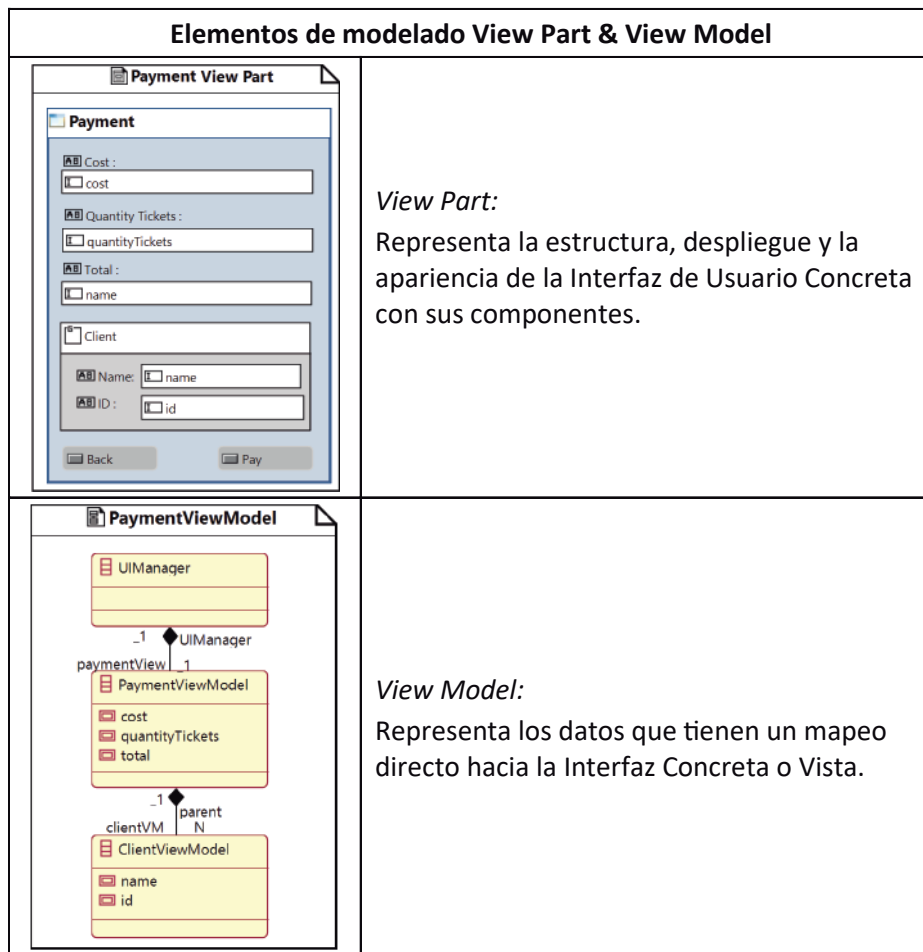


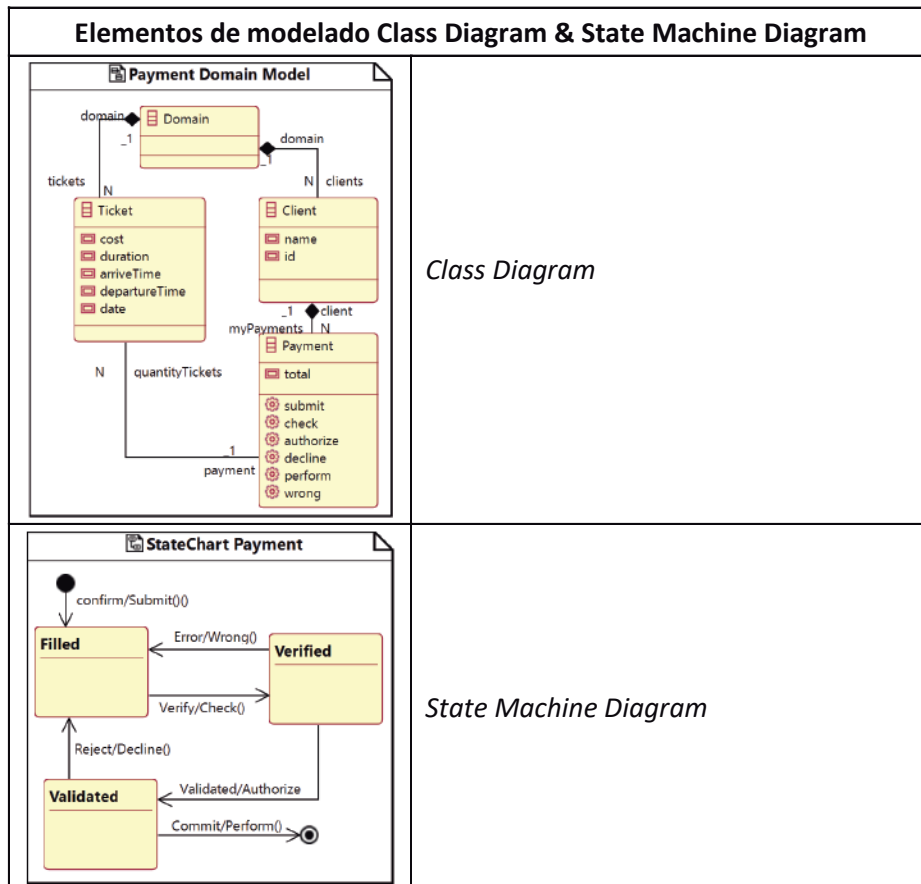
Diagrama de Clases (Class Diagram)

Es una representación de la vista estática del sistema interactivo. Los datos soportan la especificación y persistencia del sistema interactivo (Tabla D.8).

Diagrama de Máquina de Estados (State Machine Diagram)

Representan el comportamiento de las Clases que conforman el Modelo del Sistema. Cada Clase puede tener asociado uno o varios diagramas de estados (Tabla D.8).

Tabla D.8. Elementos de modelado para Class Diagram & State Machine Diagram





ANEXO E

METAMODELO DEFINIDO PARA AMBUID

En esta sección se presentan los diagramas que conforman el metamodelo de la herramienta AMBUID. El metamodelo para el desarrollo de la Interfaz de Usuario a partir del modelado de la actividad en la herramienta es representado mediante MOF en el entorno EMF de Eclipse. En las siguientes secciones se presentan cada uno de los paquetes, subpaquetes, clases y relaciones implementadas para la creación editores gráficos de la herramienta, así que cada una de estas especificaciones se crearon de acuerdo a las propiedades gráficas y conceptuales para la edición de los diferentes elementos de modelado definidos para el desarrollo de la interfaz de usuario (nivel concreto).

E.1 ESTRUCTURA DE LOS PAQUETES DE LA INTERFAZ DE USUARIO

Los principales paquetes de la Interfaz de Usuario que han sido implementados en EMF se muestran en la Figura E.1

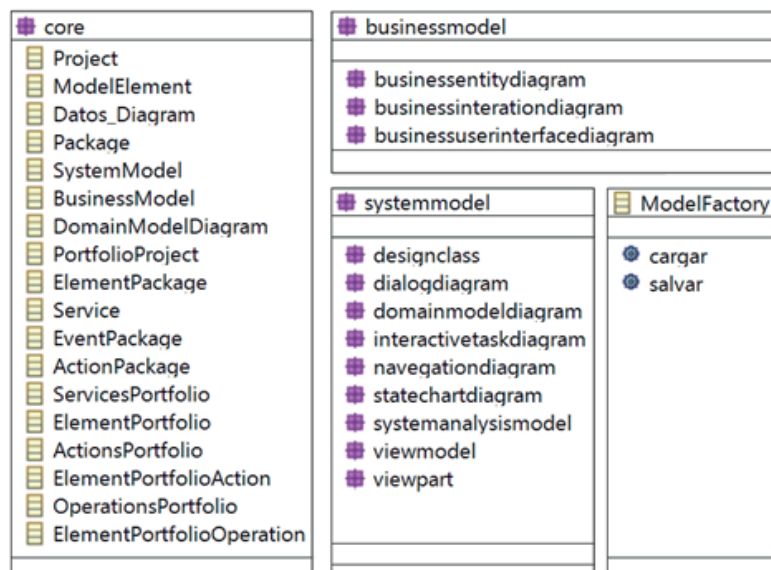


Figura E.1. Paquetes de la Interfaz de Usuario

En la Figura E.1 se ilustra el conjunto de paquetes relacionados con la Interfaz de Usuario. Se destaca los paquetes "BusinessModel" y "Systemmodel" que contiene toda la información relacionada con la sintaxis concreta de la Interfaz de Usuario para GMF. La clase "ModelFactory" se define en este nivel (raíz) para efectos de organización de los modelos EMF. De esta forma, cada que se inicia un modelo se obliga al diseñador a definir primero la información principal del sistema que se diseña. A partir de la clase "ModelFactory" se inicia la instanciación de los demás elementos de modelado que se requieran para completar el sistema.

E.2 METAMODELO DE LA INTERFAZ DE USUARIO

A continuación se presenta de manera resumida cada uno de los paquetes que conforman el metamodelo de la Interfaz de Usuario dentro de la herramienta AMBUID.

E.2.1 Paquete Core

El paquete "Core" define las principales clases que interactúan en el desarrollo de la interfaz de usuario de un sistema interactivo, permitiendo así definir la estructura compuesta del sistema al momento de especificar los modelos necesarios para su implementación. Se tiene como clase raíz "ModelFactory", desde aquí se inicia la especificación del sistema, en la Figura E.2 se puede observar que un proyecto o sistema se compone de tres elementos: a nivel de negocio se tiene el elemento "BusinessModel", a nivel de sistema se tiene el elemento "Systemmodel", a partir de estos dos elementos se crean cada uno de los diagramas pertenecientes a cada nivel. El tercer elemento "PortfolioProject" permite definir el portafolio de servicios, acciones, operaciones y eventos registrados en el sistema.

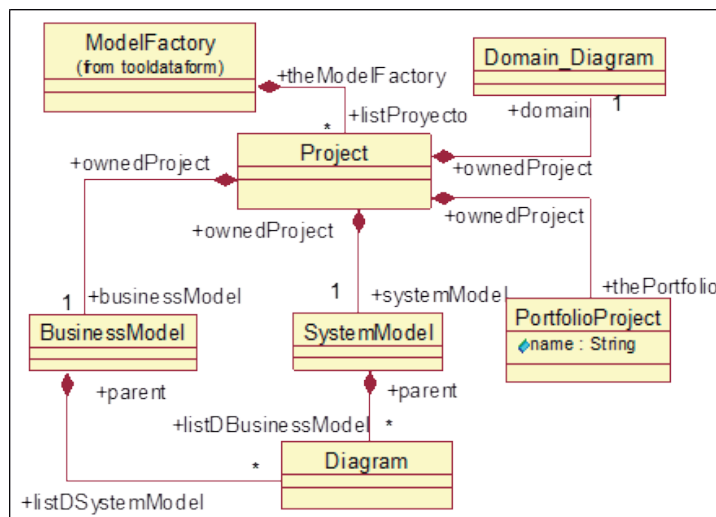


Figura E.2. Paquete Core del Metamodelo

La clase "*Domain_Diagram*" captura el modelo de datos del sistema mediante un diagrama de clases. La clase "*Diagram*" es una abstracción de cada uno de los diagramas que hacen parte del sistema, cada uno de los diagramas creados en el metamodelo son de tipo "*Diagram*".

Se ha definido dos estructuras principales para la captura de información y el modelado de un sistema interactivo de acuerdo a los dos niveles de abstracción negocio y sistema, a continuación se presenta cada una de estas estructuras.

E.2.2 Paquete Business Model

Este paquete se compone de tres subpaquetes: "*BusinessEntityDiagram*", "*BusinessInterActionDiagram*" y "*BusinessUserInterfaceDiagram*", estos paquetes hacen referencia al conjunto de diagramas pertenecientes al nivel de negocio.

En la Figura E.3 se observa las clases que hacen parte de la especificación a nivel concreto de los diagramas "*Business Entity Diagram*", "*Process Template*" y "*Business User Interface*", para la especificación de la interacción de un proceso a nivel de negocio se ha definido el elemento "Process Template", este elemento captura con detalle la información relacionada a la interacción que se presenta en un proceso de negocio.

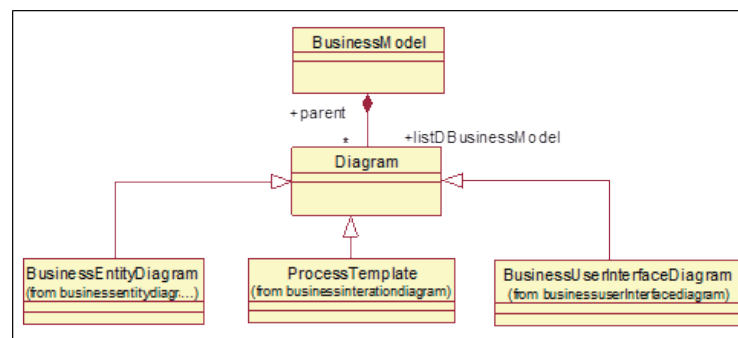


Figura E.3. Paquete Business Model

E.2.2.1 Paquete Business Entity Diagram

El paquete "*BusinessEntityDiagram*" Se centra en la descripción de las entidades del negocio y sus relaciones, en este paquete se encuentran la metaclasses necesarias para diagramar un modelo de entidades de negocio (Figura E.4).

E.2.2.2 Paquete Business Interaction Diagram

En este paquete se encuentran las metaclasses relacionadas con el modelo de interacción de negocio, entre estas metaclasses se encuentra "*Process Template*", este elemento encapsula la información relacionada con la especificación del proceso de negocio (Figura E.5).

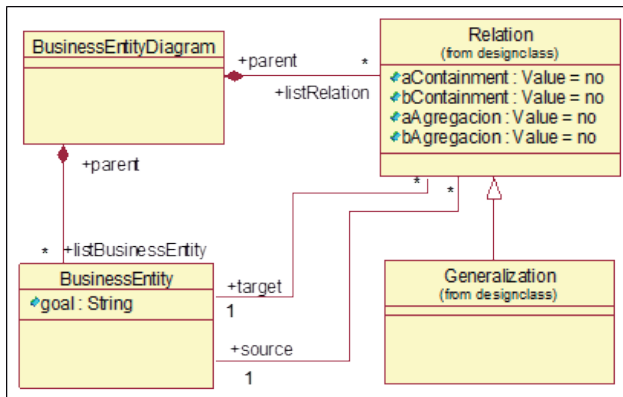
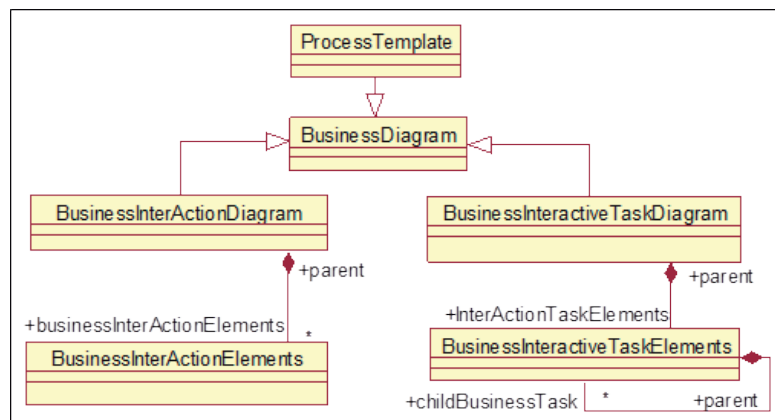


Figura E.4. Paquete Business Entity Diagram

Figura E.5. Paquete Business Interaction Diagram



En la Figura E.5 se observa que el elemento "ProcessTemplate" es un tipo de diagrama de negocio "BusinessDiagram" que a su vez es extendido por las clases "BusinessInteractionDiagram" y "BusinessInteractiveTaskDiagram", esto está definido de esta manera para tener una separación conceptual de los elementos relacionados con los objetos de negocio (BusinessInterActionElements) y los elementos de modelado que representan las tareas de negocio (BusinessInteractiveTaskElements), también permite realizar una separación a nivel grafico (Compartments en GMF) para efectos de edición de estos elementos de modelado en la herramienta.

A demás esta separación también permite realizar la implementación de estos diagramas en editores gráficos de forma independiente como se mostró en la creación de la herramienta de soporte.

A continuación se mencionan las metaclasses que intervienen en cada sub-diagrama del diagrama "ProcessTemplate".

E.2.2.2.1 Business Interaction Diagram Detallado

En el diagrama, "BusinessInterActionDiagram", se definen las metaclasses para los elementos de este modelo, como se observa en la Figura E.6.

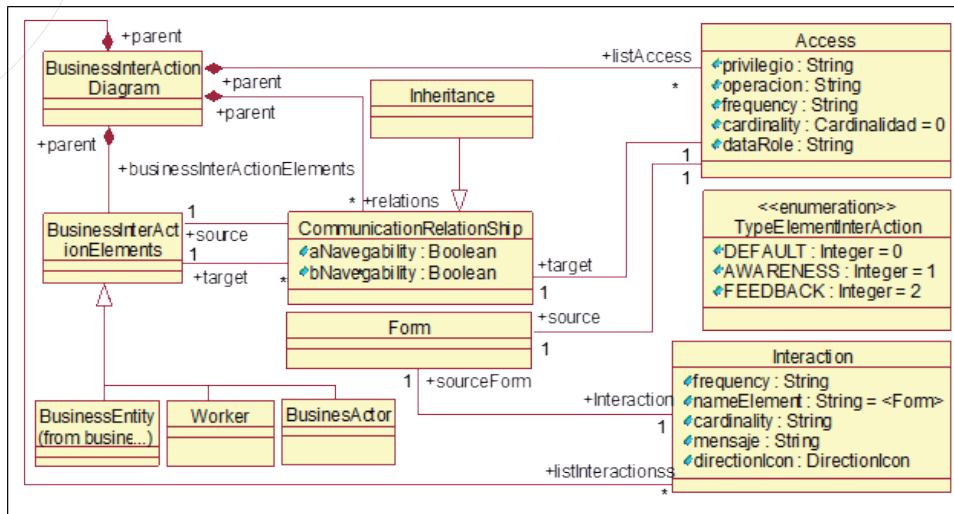


Figura E.6. Business Interaction Diagram Detallado

E.2.2.2 Business Interactive Task Diagram

La metaclass principal es "BusinessInteractiveTaskDiagram", como se observa en la Figura E.7.

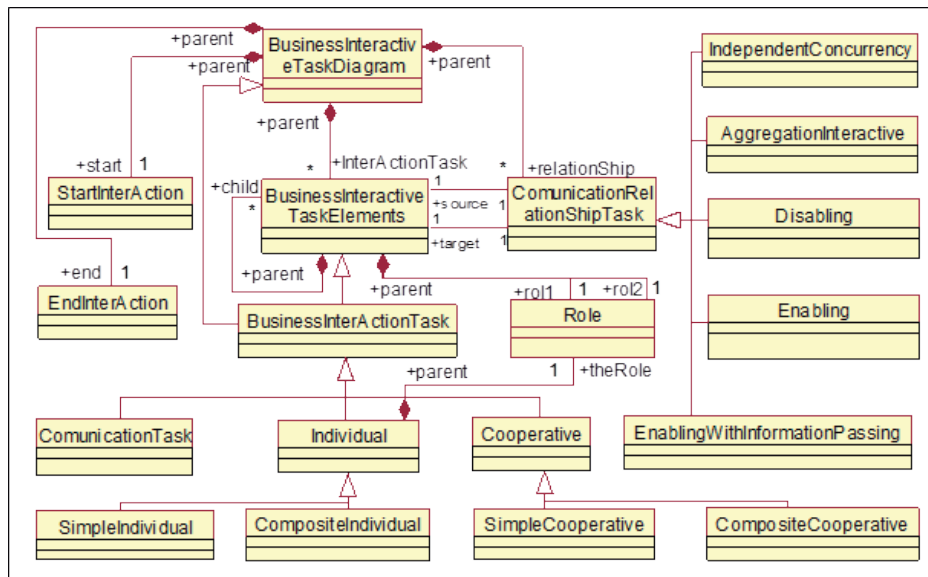


Figura E.7. Business Interactive Task Metamodel

E.2.2.3 Paquete Business User Interface

El paquete "Business User Interface Diagram" define las metaclass para describir la interfaz de usuario de negocio. Este tipo de interfaz representa una visión muy reducida de la interfaz de usuario en la que se representa los diferentes elementos basados en los formularios en papel, es decir en este modelo no se encuentran

botones o aquellos elementos gráficos "Widgets" que representen acciones (Figura E.8).

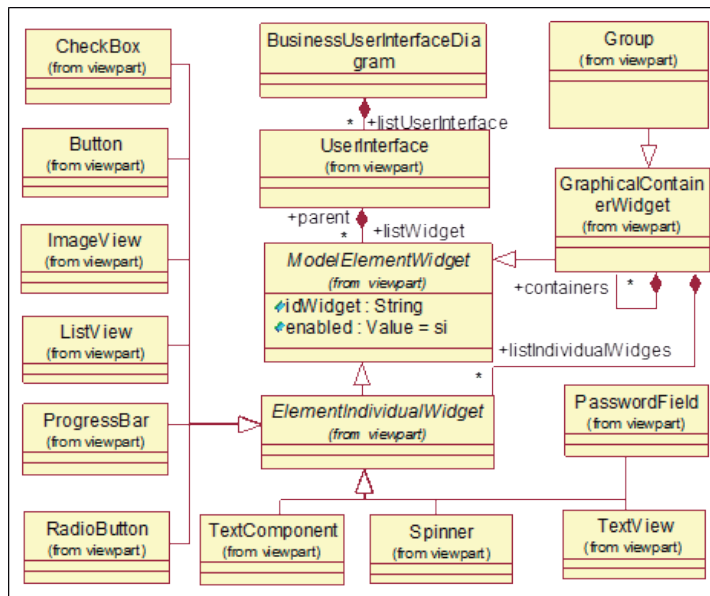


Figura E.8.. Paquete Business User Interface

E.2.3 Paquete System Model

En este paquete se encuentran las metaclasses relacionadas con los diagramas que componen el nivel de sistema, entre estas metaclasses se encuentra "Presentation Pattern", este elemento representa el patrón definido para el desarrollo de una interfaz de usuario, un patrón se compone de un modelo de tareas interactivas "InteractiveTaskDiagram", un modelo de tareas de diálogo, la interfaz de usuario concreta "UserInterfaceDiagram", el modelo de datos correspondiente a los datos de la interfaz de usuario, el modelo de datos correspondiente al dominio y las máquinas de estado asociadas a las clases de diseño para expresar su comportamiento. Por último se tiene el modelo de navegación, este modelo contiene los elementos de modelado para representar la navegación existen entre las interfaces o pantallas de un sistema. Los elementos que conforman este metamodelo y sus relaciones, se pueden observar en la Figura E.9.

El patrón "PresentationPattern", se ha definido con el fin de encapsular los modelos para el desarrollo de la interfaz de usuario de un sistema, en un patrón se modela lo relacionado solamente a una interfaz de usuario.

E.2.3.1 Paquete Design Class

El paquete "Design Class" define las metaclasses para el modelado de las clases de diseño y sus relaciones, este modelo es extendido por los modelos "ViewModel" y "DomainModelDiagram" con el fin de reutilizar su comportamiento (Figura E.10).

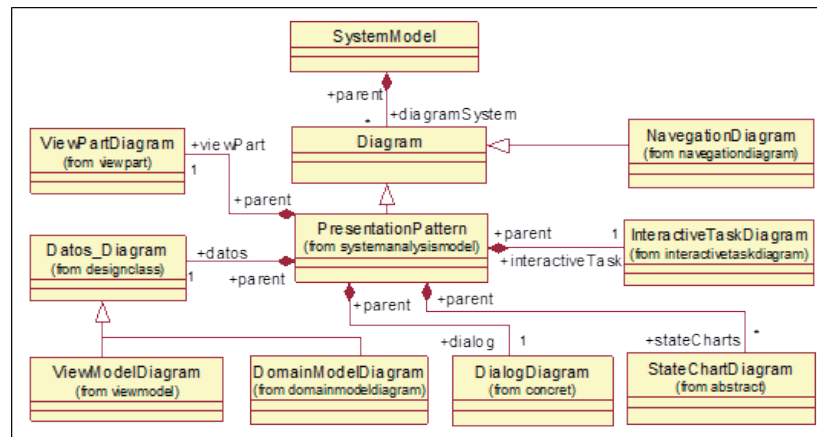


Figura E.9. Paquete System Model

Los objetos pueden describir tanto sus atributos como sus operaciones y definir sus respectivas relaciones.

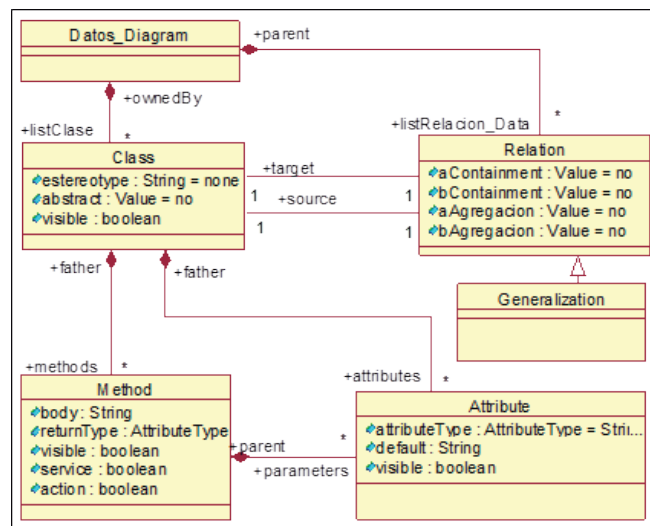


Figura E.10. Paquete Design Class

E.2.3.2 Paquete Dialog Diagram

El paquete "Dialog Diagram" define las metaclasses para el modelado de las tareas de diálogo y sus relaciones de dependencia temporal.

Un modelo de diálogo se compone de tareas de diálogo, una tarea de dialogo se compone de dos elementos: las operaciones "Dialog Operation Interaction" y un objeto de interacción "Interaction Object", el primero representa la interacción que se da en el sistema; un objeto de interacción es un componente de la interfaz "Widget" que puede ser manipulado o percibido "Interaction Type" por el usuario (por ejemplo, una ventana, un botón, una casilla de verificación, etc.), además de las tareas de diálogo este

diagrama se compone de las relaciones de dependencia temporal que se da entre tareas. (Ver Figura E.11).

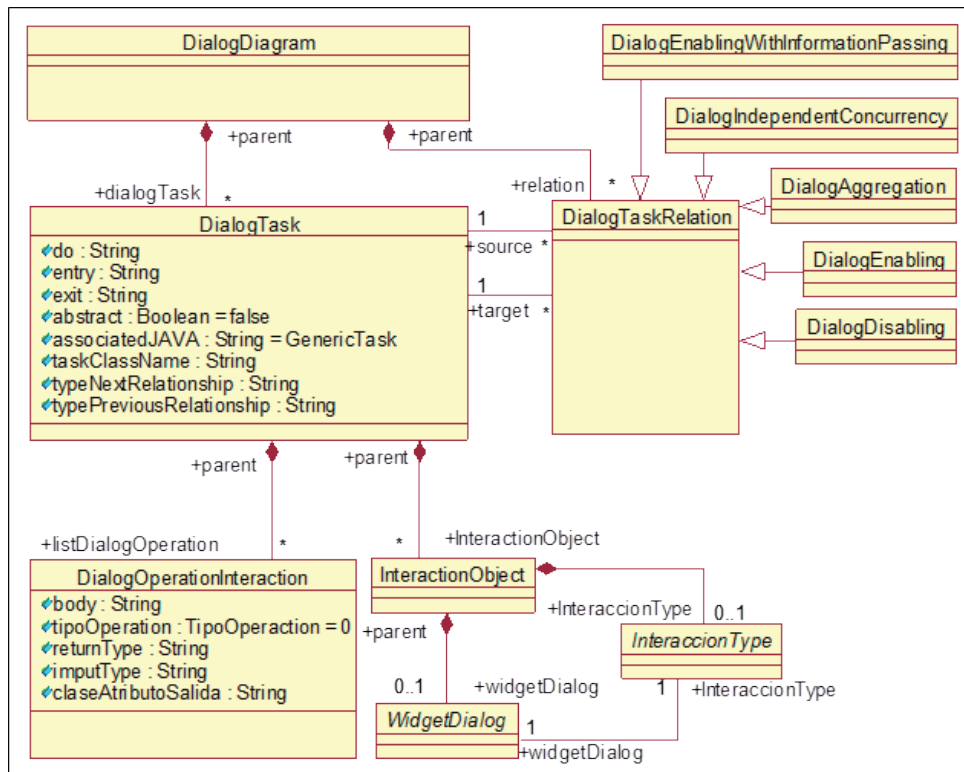


Figura E.11. Paquete Dialog Diagram

En la Figura E.12 se observa los diferentes widgets que hacen parte del objeto de interacción. En la figura E.13, se observa los diferentes tipos de interacción que hacen parte de la tarea de diálogo.

E.2.3.4 Paquete Domain Model Diagram

El paquete "DomainModel Diagram" define las metaclasses para describir un modelo de dominio. En este paquete se encuentra la metaclassa "DomianModelDiagram" que es un tipo de diagrama de datos, esto para reutilizar comportamiento ya que los datos del dominio se representan mediante clases y relaciones entre ellas (Figura E.14).

E.2.3.5 Paquete Interactive Task Diagram

El paquete "Interactive Task Diagram" define las metaclasses para el modelado de las tareas interactivas y sus relaciones para representar los aspectos interactivos del sistema (Figura E.15).

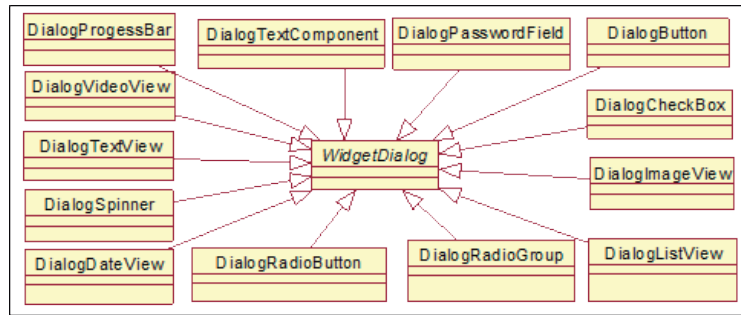


Figura E.12. Widgets de una tarea de diálogo

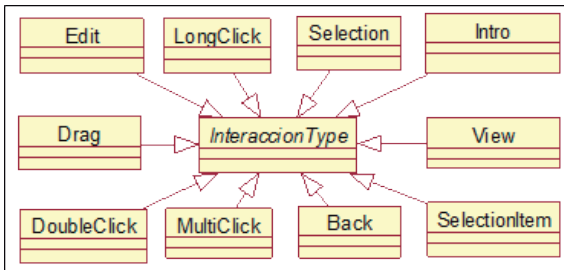


Figura E.13. Tipos de interacción de una tarea de diálogo

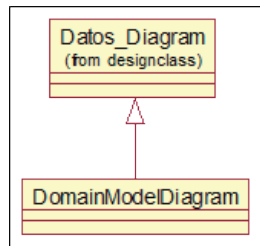


Figura E.14. Paquete Domain Model Diagram

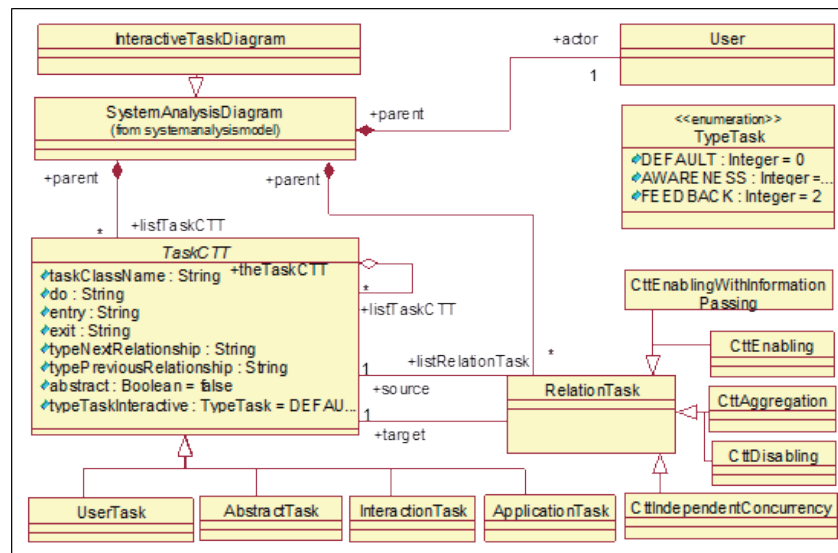


Figura E.15. Paquete Interactive Task Diagram

E.2.3.6 Paquete Navigation Diagram

El paquete "Navigation Diagram" define las metaclasses para el modelado de la navegación entre las interfaz de usuario del sistema (Figura E.16).

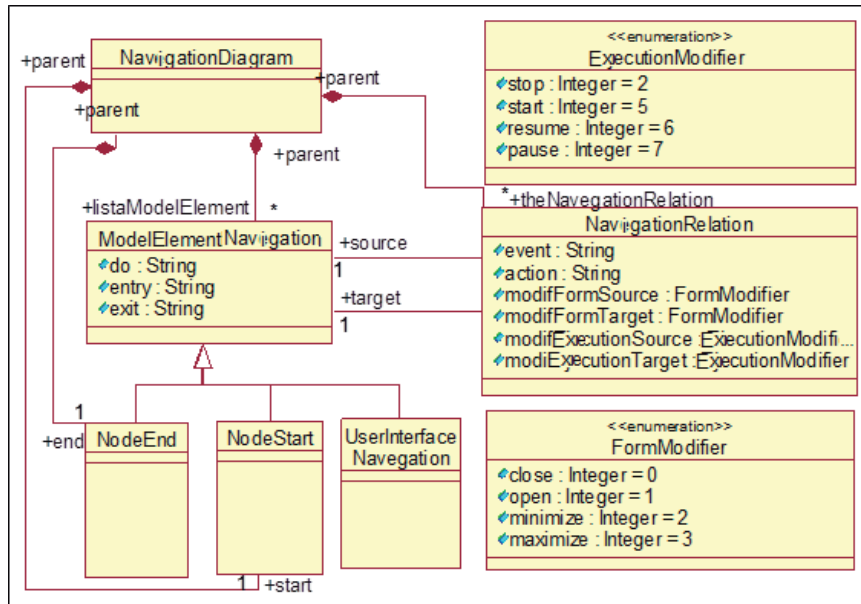


Figura E.16. Paquete Navigation Diagram

E.2.3.7 Paquete Statechart Diagram

Este paquete tiene como objetivo la definición de la estructura (nivel concreto o notacional) de las máquinas de estado. Estos diagramas representan el comportamiento de las clases y casos de uso, respectivamente. La sincronización de las máquinas de estado se lleva a cabo mediante los eventos. Un evento está capacitado para disparar un método por intermedio de la operación asociada. Similarmente, un evento puede desencadenar una serie de acciones las cuales a su vez puede activar eventos (Ver Figura E.17).

E.2.3.8 Paquete View Model Diagram

El paquete "View Model Diagram" define las metaclasses para el modelado del modelo de datos asociado a la interfaz de usuario, este modelo representa la persistencia de los datos de una interfaz. La meta clase "ViewmodelDiagram" hereda el comportamiento del modelo de diseño (Figura E.18).

E.2.3.9 Paquete View Part Diagram

El paquete "View Part Diagram" define las metaclasses para el modelado de la interfaz de usuario concreta (Figura E.19). En la (Figura E.20), se muestran los diferentes widgets que hace parte del modelo de la view Part o interfaz de usuario.

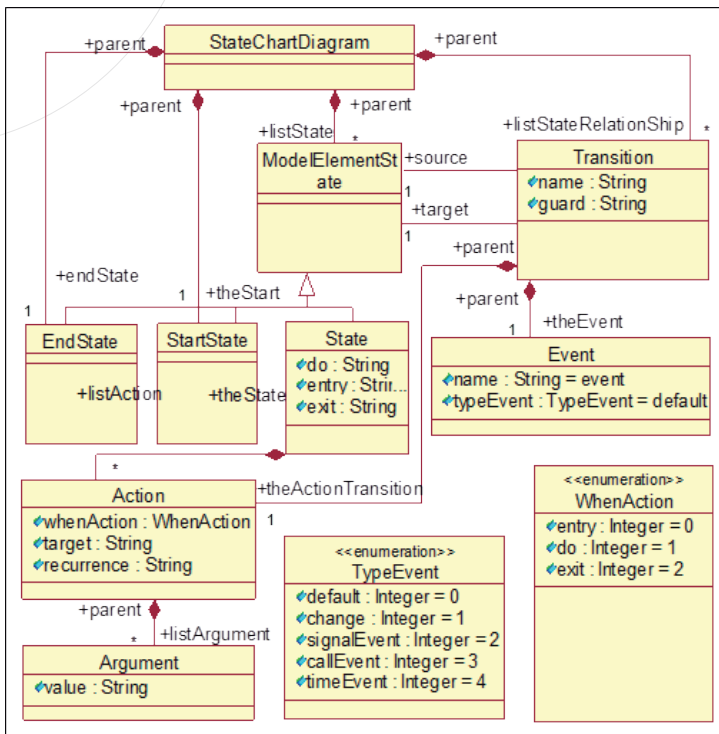


Figura E.17. Paquete StateChart Diagram

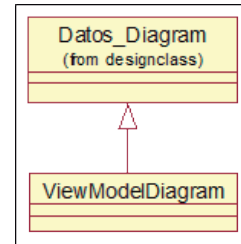


Figura E.18. Paquete ViewModel Diagram

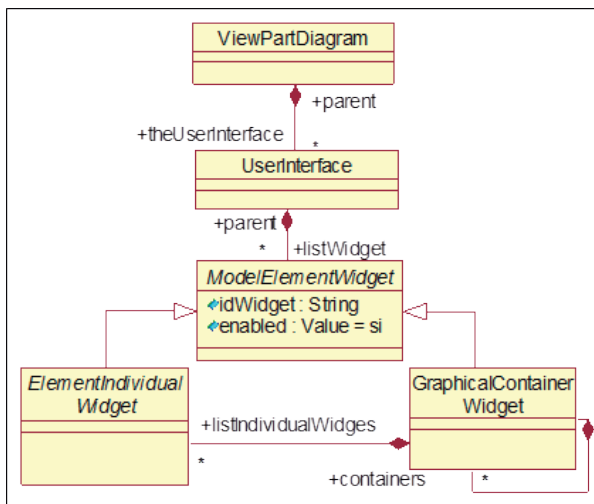
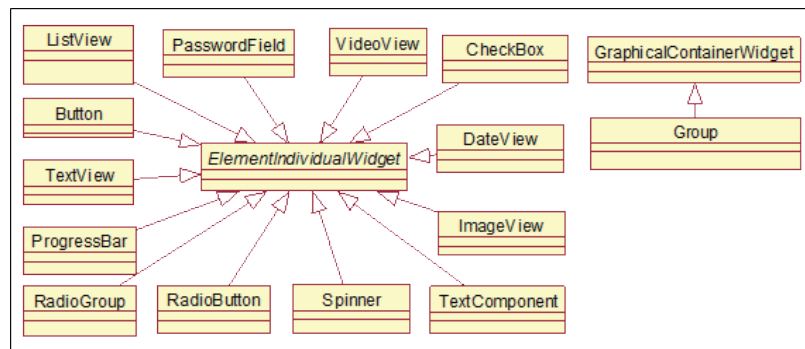


Figura E.19. Paquete ViewPart Diagram

Figura E.20. Widgets de la ViewPart





ANEXO F

MODELO DE DISEÑO DE EJECUCIÓN DE MÁQUINAS DE ESTADO

F.1 MODELO DE CONCURRENCIA

UML Semantics, soporta la implementación de máquinas de estado. Generalmente, las máquinas de estado representan el comportamiento de una clase. Es posible que una clase tenga asociada más de una máquina de estado, y en una aplicación se modelan gran variedad de clases. Cada máquina de estado debe estar activa, esperando por los eventos que provocan una conmutación de estado. Por esta razón el mecanismo que inicia la ejecución de cada máquina de estado (`Run()`) debe manejarse como un *thread*.

Tal como lo expresa el metamodelo de UML, cada estado Top posee una lista de eventos diferidos. Dicha lista es administrada por una clase que no está especificada en el metamodelo (**EventManager**) pero es necesario adicionarla al metamodelo como interfaz entre el kernel EMTRSAI y las metaclasses especificadas en el metamodelo de UML. El manejador de eventos está dentro del estado Top, porque él es el encargado de despachar los eventos. Esto lo hace a través de las transiciones, las cuales están asociadas con cada estado y con la máquina.

El mecanismo de despacho de eventos (`RunSteps()`) debe ser un proceso, con el fin de que se revisen constantemente los eventos pendientes. La Figura F.1 muestra la estructura de la lista de eventos diferidos y el manejador de eventos que contiene el estado Top.

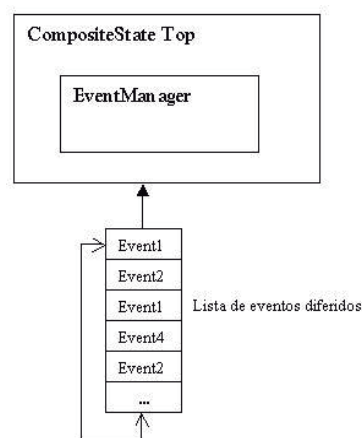


Figura F.1. Esquema de eventos diferidos

Si una máquina de estado está activa, uno de sus estados internos debe estar activo también. Mientras que uno de los subestados de la máquina está realizando sus

actividades internas, se debe estar chequeando la ocurrencia de un evento que dispare alguna transición de la máquina. Si ocurre algún evento por el que esté esperando el estado corriente para conmutar, debe detenerse la ejecución de sus actividades y cambiar de estado. Esto demuestra la existencia de dos procesos concurrentes. Uno es el que revisa la ocurrencia de los eventos constantemente, nombrado anteriormente, el otro es el proceso de activación de un estado (`Activate()`).

F.1.1 Diagrama de tareas

La semántica de UML expresa que cada máquina de estado está compuesta por un estado top, el cual debe ser de tipo `Composite`. La relación mostrada entre **StateMachine** y **CompositeState** en la Figura F2 expresa que la máquina de estado activa al estado top al iniciar su ejecución, por lo tanto ella es la que monta el proceso de `Activate()` del estado top.

Cada estado compuesto que sea top de una máquina contiene un proceso `RunSteps()`, con el que se avanza un paso en la ejecución de la máquina. Este proceso es montado por el estado compuesto, una vez que se ponga activo.

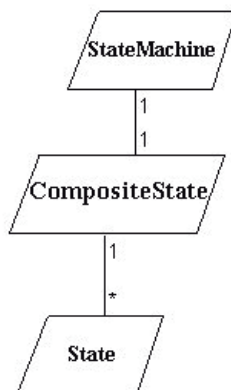


Figura F.2. Diagrama de tareas del sistema

Cada máquina de estado contiene un proceso `Run()`, que es el que inicia su ejecución. El constructor de la máquina es el responsable de montar dicho proceso. Todos los estados de la máquina contienen un proceso `Activate()`, que es el que ejecuta las acciones de entrada, de salida y las actividades internas del estado. El estado top es el responsable de montar dicho proceso. La máquina de estado activa al estado top al iniciar su ejecución, por lo tanto ella es la que monta el proceso de `Activate()` del Estado Top. El número de procesos montados en UML Semantics, depende del número de máquinas de estado que se construyan en la aplicación, y de cuántos estados contenga cada máquina

Se dispone de con un Ejecutivo Multitarea en Tiempo Real para que sea posible el manejo de procesos concurrentes y de recursos, a la hora de implementar varias máquinas de estado. Pensando en la reutilización del código, y en el uso de *Semantics.lib* sobre otros sistemas operativos, se ha decidido que no se deben hacer llamadas a funciones específicas del kernel. En lugar de ello, se debe crear una capa de interfaz, es decir, una clase que provea las primitivas que contiene un kernel en tiempo

real, como `AdicionarProcesos()`, `IniciarEjecución()`, y que sea posible reemplazar el componente kernel cuando se considere necesario.

F.1.2 Diagrama de estados concurrentes

Es posible que una máquina de estado esté corriendo (*Running*) ó que no haya iniciado su ejecución (*Stopped*), (Figura F.3 (a)). En caso de que esté corriendo, activará su estado Top, y esperará hasta que se llegue al estado final para desactivarlo. Al activarse el estado Top, se inicia la ejecución de la máquina de estado mostrada en la Figura F.3 (b). Cuando la máquina llega al estado `ProcessingEvent`, se activa un subestado de la máquina si el evento procesado es válido para que se conmute de estado, además de ejecutarse otras actividades. La máquina de estado correspondiente a la activación de un subestado de la máquina se muestra en la Figura F.3 (c). Las máquinas de estado *RunningSteps* y *DoingActions* se ejecutan concurrentemente, ya que mientras que el estado top de la máquina se encuentra esperando por nuevos eventos, uno de sus subestados se encuentra activo ejecutando sus acciones y actividades internas. Es posible que el comportamiento de una clase se encuentre modelado por más de una máquina de estado. En este caso, se tendrían varios estados *Running* conmutando en paralelo en la aplicación.

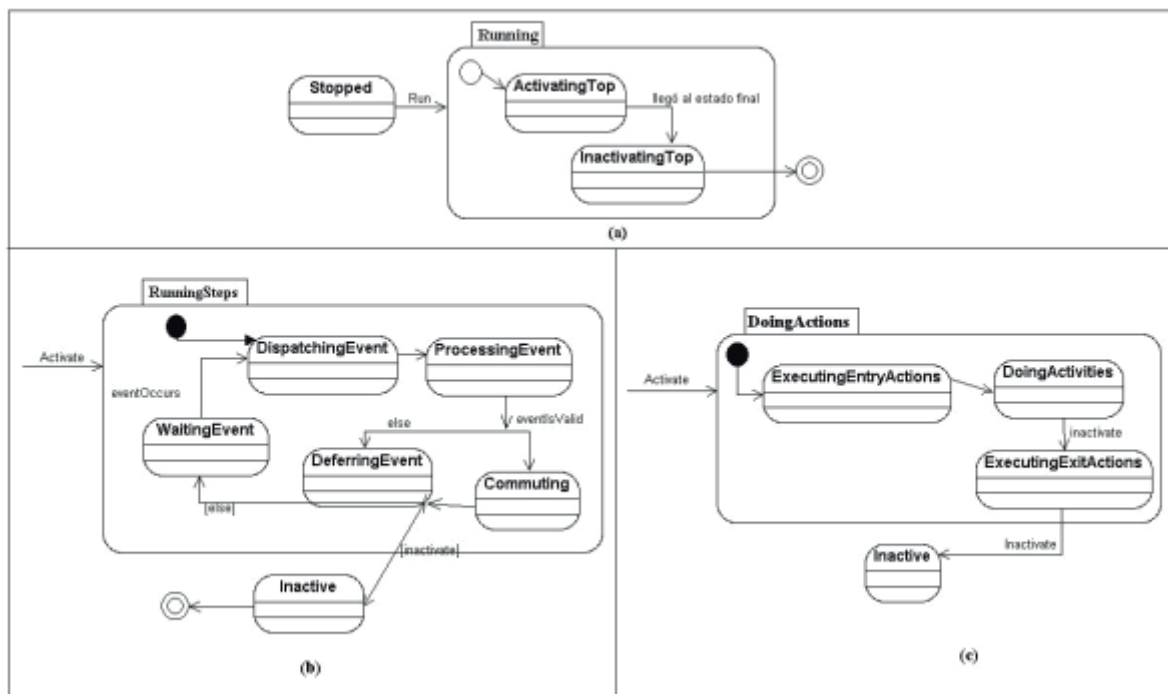


Figura F.3. Diagrama de estados concurrentes

