



Universidad Castilla-La Mancha
Escuela Superior de Ingeniería Informática
Departamento de Sistemas Informáticos

Programa Oficial de Postgrado en
Tecnologías Informáticas Avanzadas

Trabajo Fin de Máster

**ANALISIS Y MODELADO DE TAREAS EN LOS NUEVOS
PARADIGMAS DE INTERACCION HOMBRE-MAQUINA**

Julio 2012

Alumno Jaslin Fernando Flores Avilez
Director Dr. Francisco Montero Simarro
Co-Director Dr. Víctor López-Jaquero

Contenidos

Índice de figuras	v
Índice tablas	vii
Resumen	ix
Capítulo 1	11
Asignaturas cursadas	11
1.1 Generación de Documentos Científicos en Informática	11
1.1.1 Descripción de la asignatura.....	11
1.1.2 Trabajo Realizado.....	12
1.2 Introducción a la Programación de Arquitecturas de Altas Prestaciones.....	13
1.2.1 Descripción de la asignatura.....	13
1.2.2 Trabajo realizado	14
1.3 Sistemas Inteligentes Aplicados a Internet	16
1.3.1 Descripción de la asignatura.....	16
1.3.2 Trabajo Realizado.....	16
1.4 Calidad de Interfaces de Usuario: Desarrollo Avanzado.....	17
1.4.1 Descripción de la asignatura.....	17
1.4.2 Trabajo realizado	17
1.5 Tecnología de Software Orientada a Objetos	18
1.5.1 Descripción de la asignatura.....	18
1.5.2 Trabajo realizado	19
1.6 Programación Internet con Lenguajes Declarativos Multiparadigma	19
1.6.1 Descripción de la asignatura.....	19
1.6.2 Trabajo realizado.	20
Capítulo 2	21
Estado del Arte	21
2.1 Introducción, contexto y alcance de la investigación: Nuevos paradigmas de interacción.....	21
2.2 Interacción hombre máquina.....	23
2.2.1 Estilos de interacción hombre-máquina	23
2.2.2 Paradigmas y modalidades de interacción.	26
2.3 Desarrollo de interfaces de usuario.....	27
2.4 Análisis y modelado de tareas.....	30
2.4.1 Definiciones previas relacionadas con modelado de tareas	31

2.4.2	Técnicas de análisis y modelado de tareas basadas en UML	32
2.4.3	Análisis jerárquico de tareas / Hierarchical Task Analysis.....	33
2.4.4	GOMS (Goals, Operators, Methods, Selection rules).....	36
2.4.5	La notación ConcurTaskTrees	38
2.4.6	UML y sus facilidades para la especificación de tareas.....	41
2.5	Discusión de las técnicas y notaciones de análisis y modelado de tareas	41
2.5.1	Nuevos paradigmas de interacción vs paradigmas tradicionales de interacción	42
2.5.2	Diferencia sintáctica (textual vs grafica)	42
2.5.3	Diferentes niveles de formalidad.....	43
2.5.4	Diferente conjunto de operadores	43
2.5.5	Similitudes entre las notaciones de análisis y modelado de tareas.....	43
2.6	Trabajos relacionados.....	44
2.7	Conclusiones	47
Capítulo 3	49
Trabajo de Investigación: Evaluando las técnicas de análisis y modelado de tareas considerando distintos criterios de interacción persona-ordenador		49
3.1	Análisis empírico de técnicas de análisis y modelado de tareas en diferentes tipos de plataforma y modalidad	49
3.2	Evaluación de técnicas de análisis y modelado de tareas en paradigmas post-WIMP	51
3.2.1	Evaluación de las técnicas y notaciones de análisis y modelado de tareas	51
3.3	Análisis global de los resultados.....	64
3.4	Conclusiones	65
Capítulo 4	67
Anteproyecto de Tesis.....		67
4.1	Introducción	67
4.2	Dominio del problema	67
4.3	Propuesta de tesis	68
4.4	Planificación de actividades	70
4.5	Cronograma.....	72
Capítulo 5	75
Currículo Vitae		75
5.1	Titulación académica	75
5.2	Becas Disfrutadas	75
5.3	Otros Méritos	75
5.3.1	Experiencia profesional.....	75

5.3.2	Otros cursos.....	76
Referencias.....		77

Índice de figuras

FIGURA 1-1. GRÁFICO TIEMPOS EMPLEADOS/MATRIZ CON MPI Y BLAS	14
FIGURA 1-2. GRÁFICO TIEMPOS EMPLEADOS/MATRIZ MPI Y PBLAS	15
FIGURA 2-1. ALCANCE Y DIMENSIÓN DE LA DISCIPLINA INTERACCIÓN PERSONA-ORDENADOR	22
FIGURA 2-2. GRAFICA DE INTERFAZ POR LÍNEA DE ÓRDENES	24
FIGURA 2-3 EJEMPLO DE INTERACCIÓN BASADA EN CUMPLIMENTACIÓN DE FORMULARIOS.....	24
FIGURA 2-4. EJEMPLO DE INTERACCIÓN BASADA EN MENÚS	25
FIGURA 2-5 EJEMPLO DE INTERACCIÓN BASADA EN LA MANIPULACIÓN DIRECTA	25
FIGURA 2-6. ESQUEMA GENERAL ASOCIADO A UN ENTORNO DE DESARROLLO DE INTERFACES DE USUARIO BASADO EN MODELOS	27
FIGURA 2-7. MODELOS Y RELACIONES ENTRE ELLOS DEFINIDOS EN UN Mb-UIDE.....	28
FIGURA 2-8. CAMELEON FRAMEWORK	29
FIGURA 2-9 DIAGRAMA DE ACTIVIDADES DE CONTROL DE CRÉDITO	33
FIGURA 2-10. JERARQUÍA DE METAS PARA UN ATM	35
FIGURA 2-11. MODELO CTT DE UN ATM.....	41
FIGURA 2-12 RELACIÓN ENTRE MODELOS EN UN ENTORNO CSCW	45
FIGURA 3-1. MÉTODO INVESTIGACIÓN-ACCIÓN Y DESMET	50
FIGURA 3-2 NIVEL DE IMPORTANCIA OBTENIDO SEGÚN EL GRUPO DE EXPERTOS	57
FIGURA 3-3 DISCIPLINAS CON LA QUE GUARDAN MÁS AFINIDAD LOS EXPERTOS	57
FIGURA 3-4 NOTACIONES DE ANÁLISIS Y MODELADO DE TAREAS CON MÁS EXPERIENCIA	58
FIGURA 3-5 TIEMPO DE EXPERIENCIA EN NOTACIONES DE MODELADO DE TAREAS	58
FIGURA 3-6 RESULTADO DE EVALUACIÓN DE TÉCNICAS/NOTACIONES	65
FIGURA 3-7 RESULTADO DE COMPARATIVA EN PORCENTAJE DE CADA CARACTERÍSTICA.....	65
FIGURA 4-1 MODELOS DEL FRAMEWORK CAMELEON PARA FASES DE DISEÑO Y EJECUCIÓN	68
FIGURA 4-2 MODELOS EN MDD.....	70
FIGURA 4-3 CRONOGRAMA DE ACTIVIDADES	73

Índice tablas

TABLA 2-1. DEFINIR DATOS EN GOMS.....	36
TABLA 2-2. DEFINIR OPERADORES EN GOMS	37
TABLA 2-3. MÉTODO EN MODELO GOMS	37
TABLA 2-4. REGLAS DE SELECCIÓN EN MODELO GOMS	38
TABLA 2-5. EJEMPLO DE UN MODELO GOMS	38
TABLA 2-6. OPERADORES TEMPORALES EN CTT	40
TABLA 2-7 SIMILITUDES Y DIFERENCIAS ENTRE TÉCNICAS DE ANÁLISIS Y MODELADO DE TAREAS	44
TABLA 2-8 RESUMEN DE TÉCNICAS EN APORTACIONES	47
TABLA 3-1 FASES Y ACCIONES DEL MÉTODO INVESTIGACIÓN-ACCIÓN	50
TABLA 3-2 LISTA DE CARACTERÍSTICAS DE UNA TÉCNICA/NOTACIÓN GENERAL DE ANÁLISIS Y MODELADO DE TAREAS	53
TABLA 3-3 LISTA DE GRADO DE IMPORTANCIA USANDO DESMET	55
TABLA 3-4 NIVEL DE IMPORTANCIA DE LAS CARACTERÍSTICAS DE TÉCNICA/NOTACIÓN GENERAL DE ANÁLISIS Y MODELADO DE TAREAS.....	56
TABLA 3-5 ESCALA DE JUICIO PARA EVALUAR EL SOPORTE DE LAS CARACTERÍSTICAS SEGÚN DESMET	59
TABLA 3-6 EVALUACIÓN PARA TÉCNICAS BASADAS EN UML	60
TABLA 3-7 EVALUACIÓN DE HTA	61
TABLA 3-8 EVALUACIÓN DE GOMS.....	63
TABLA 3-9 EVALUACIÓN DE CTT	63

Resumen

En la actualidad, la mayoría de los problemas que se presentan durante el desarrollo de interfaces de usuarios tienen que ver con la necesidad de considerar filosofías de Diseño Centradas en los Usuarios y la falta de herramientas que soporten todas las etapas importantes del desarrollo de aplicaciones. Dentro de estas etapas se encuentran las que van desde el análisis de los requisitos, que es lo que sirve para empezar el modelado de cualquier tipo de sistema, hasta el desarrollo y la implementación de las aplicaciones.

Este trabajo fin de Máster tiene la intención de contribuir a la mejora de las técnicas y entornos de elaboración de interfaces de usuario. Entre ellos, los entornos de desarrollo de interfaces de usuario basados en modelos (Mb-UIDE) que son las técnicas que vienen utilizándose con éxito durante las últimas dos décadas. En ellas, las actividades de análisis y modelado aparecen de manera constante y reiterada, pero en la actualidad perviven diferentes notaciones que se utilizan en las distintas propuestas Mb-UIDE disponibles. En este trabajo pretendemos, en primera instancia identificar, analizar y comparar dichas notaciones, metodologías y herramientas y, en segundo lugar, proporcionar un mecanismo que facilite su elección en función de las interfaces de usuario que se pretendan desarrollar.

Las actividades de análisis y modelado siempre han estado y están presentes dentro de las actividades de desarrollo de interfaces de usuario. Independientemente de que nos acerquemos a estas actividades desde diferentes disciplinas. Ejemplo de esas disciplinas son tanto la Ingeniería del Software como la Interacción Persona-Ordenador. Por ello, las alternativas y propuestas disponibles son muy variadas. De hecho en la propia Escuela Superior de Ingeniería en Informática de Albacete se han desarrollado diferentes tesis doctorales en la última década y, aunque en todas ellas se usa el concepto de Mb-UIDE, en ellas se han utilizado distintas notaciones que soportan las actividades de análisis y modelado sin una justificación adecuada de la elección de la notación/lenguaje finalmente utilizada.

Las especificaciones relacionadas con las actividades de análisis y modelado, en ocasiones, son independientes de la plataforma y de la modalidad de interacción. En este sentido, distintos trabajos de investigación tanto europeos como americanos han considerado diferentes notaciones para soportar el análisis y el modelado pero, en ellos, tampoco no se ha justificado la elección ni se ha desarrollado ningún trabajo que ayude a seleccionar o determinar qué notación utilizar y cuándo utilizarla. Por ello, el trabajo fin de Máster que se presenta en este documento está justificado y se centra en un tipo concreto de esas notaciones; las dedicadas al análisis y modelado de tareas.

Como trabajo futuro del trabajo realizado hasta el momento, se propone considerar el mecanismo de selección propuesto para potenciar las propuestas Mb-UIDE y soportar el desarrollo de las nuevos interfaces de usuario que están surgiendo y desarrollándose en la actualidad, analizando, de una manera crítica, las herramientas de soporte (metodologías, notaciones y herramientas) disponibles.

El presente documento se organiza como sigue. En el primer capítulo se enumerarán y describirán las asignaturas cursadas en el periodo de formación académica asociado a los estudios de doctorado. El capítulo segundo presenta el estado del arte relacionado con la

propuesta de investigación que se propone. El capítulo tercero documenta el trabajo de investigación realizado, que consiste en evaluar algunas de las diferentes notaciones de modelado de tareas disponibles a partir de una serie de criterios considerados. El capítulo cuarto presenta la propuesta de anteproyecto de tesis doctoral y la planificación que se llevará a cabo para realizar este cometido. Por último, el quinto capítulo presenta una breve descripción del CV del autor de este documento.

Capítulo 1

Asignaturas Cursadas

El presente capítulo realiza una descripción y una recopilación de las asignaturas cursadas y de los trabajos realizados en cada una de ellas. Las asignaturas se cursaron en el periodo de formación académica del Doctorado en Tecnologías Informáticas Avanzadas, las asignaturas corresponden al curso académico 2011/2012.

Los cursos se llevaron a cabo en la Escuela Superior de Ingeniería Informática en el Campus Universitario de Albacete en la comunidad de Castilla-La Mancha.

Algunas de las asignaturas cursadas tienen una relación directa con la línea de investigación y el trabajo fin de Máster que propongo. Durante este primer período de formación académica se han cursado las siguientes asignaturas:

- Generación de Documentos Científicos en Informática
- Introducción a la Programación de Arquitecturas de Altas Prestaciones
- Sistemas Inteligentes Aplicados a Internet
- Calidad de Interfaces de Usuarios: Desarrollo Avanzado
- Programación Internet con Lenguajes Declarativos Multiparadigma
- Tecnología de Software Orientada a Objetos

Para describir las actividades realizadas en cada una de las asignaturas, que se han cursado, se utilizarán dos secciones. En primer lugar se describirá la asignatura, haciendo referencia a sus profesores responsables y comentando los principales conceptos presentados en ellas. Después, en la sección trabajo realizado, se describirá brevemente el trabajo realizado para superar cada asignatura.

1.1 Generación de Documentos Científicos en Informática

1.1.1 Descripción de la asignatura

La asignatura Generación de Documentos Científicos está orientada a la metodología de la investigación científica brindándonos una serie de herramientas y pautas para poder llevar a cabo el proceso de investigación de forma efectiva y eficiente. La labor investigadora implica el planteamiento de una serie de cuestiones sobre un evento, contexto o desafío y tratar de dar respuestas a tales cuestiones.

La asignatura fue presentada por los doctores José Antonio Gámez, Francisco Parreño y Luis de la Ossa. La asignatura está dividida en tres bloques.

En el primer bloque impartido por el Dr. D. José Antonio Gámez se abordaron conceptos generales como definiciones tales como *conocimiento*, *tecnología*, *investigar*, entre otros. Además, se identificaron criterios sobre cómo llevar a cabo el método científico; garantizar la reproducibilidad de un experimento, demostrar si es verdadero o no el mismo, etc. En este

bloque se realizó un trabajo relacionado con el índice h de J. E. Hirsch y la nueva propuesta de índice \bar{h} (h -barra) redefinido por el mismo autor. Este último índice, a diferencia del primero caracteriza de manera única un documento como la pertenencia o no pertenencia al \bar{h} núcleo de sus autores.

En el segundo bloque impartido por el Dr. D. Francisco Parreño se nos presentaron herramientas para poder llevar a cabo una investigación científica. En este bloque se presentó que las herramientas fundamentales para llevar a cabo una investigación objetiva son las herramientas estadísticas ya que ofrecen métodos para responder a una serie de hipótesis y poder dar por verdadera o falsa una de estas previamente definidas en la concepción del problema que se quiere abordar. El profesor Parreño nos presentó una herramienta automatizada o aplicación de software denominada 'R' (The R Project for Statistical Computing 2012), la cual realiza una serie de pruebas o test estadísticos, paramétricos y no paramétricos, para poder evaluar la veracidad de las hipótesis planteadas en nuestras investigaciones, además la herramienta facilita la construcción de gráficas para poder mostrar e interpretar mejor los resultados arrojados por las pruebas estadísticas.

Por último, el tercer bloque impartido por el Dr. D. Luis de la Ossa presentó una herramienta muy utilizada por la comunidad científica para la elaboración de todo tipo de documentos e informes científicos como artículos, libros, etc. El editor de textos presentado se denomina 'LATEX' (LaTeX 2012) y ofrece facilidades de edición de ecuaciones matemáticas de forma sencilla y precisa.

1.1.2 Trabajo Realizado

En la asignatura Generación de Documentos Científicos en Informática el trabajo realizado consistió en la realización de un conjunto de pruebas estadísticas, tanto paramétricas como no paramétricas, de un problema dado.

El test paramétrico consistió en realizar un Análisis de Varianza según (M, M and Chía 2006) (ANOVA en inglés) de un conjunto de datos que fueron extraídos de la asignatura de Introducción a la Programación de Arquitecturas de Altas Prestaciones. En ella se desarrollaron programas utilizando el lenguaje de programación 'C', sobre el producto de dos matrices, tomando como factor de análisis o de influencia el orden en el que se accedía a las columnas, tales ordenes fueron las siguientes: "ijk", "jki", "kji".

El test no paramétrico (M. M. López Sánchez M 2006) consistió en la utilización de la prueba de Kruskal-Wallis, un método no paramétrico para probar si un grupo de datos proviene de la misma población o no. Esencialmente, la prueba de Kruskal-Wallis es similar a un análisis ANOVA con los datos reemplazados por categorías. La utilidad de este tipo de test no paramétrico está en poder utilizarlo incluso cuando no se cuenta con suficientes datos a diferencia del Análisis de Varianza. Los datos de la prueba de Kruskal-Wallis se tomaron de la ejecución de un programa escrito en el lenguaje de programación 'C'. Este programa consistía en hacer uso de una Interfaz de Paso de Mensajes (MPI) muy usada en programación paralela, ya que permite que se pueda distribuir el cálculo de diferentes tareas de manera parcial, haciendo uso de varios procesos y tratando de aprovechar al máximo el rendimiento de los equipos con los que se cuenta. También se utilizó Basic Lineal Algebra (BLAS). BLAS, una

librería muy usada ya que cuenta con subrutinas que agilizan de manera más eficiente los cálculos matemáticos.

El test no paramétrico consistió en saber si los factores orden y número de nodos, que se empleaba al realizar los cálculos en un clúster, influían en el producto de dos matrices.

Las conclusiones alcanzadas tanto en el test paramétrico como en el no paramétrico fueron las siguientes:

- El Test paramétrico, ANOVA, permitió comprobar la hipótesis nula de si el factor orden afecta al tiempo empleado para realizar los cálculos del producto $C=A*B$. Fruto de este test se comprobó que, efectivamente, no afecta.
- El test no paramétrico, KRUSKAL-WALLIS, también permitió comprobar si otro factor, el número de nodos afectaba al tiempo empleado en los cálculos. El resultado confirmó la hipótesis nula y ratificó que este factor no afecta.

Fruto de las pruebas realizadas se llegó a la aceptación de las hipótesis nulas previamente planteadas.

1.2 Introducción a la Programación de Arquitecturas de Altas Prestaciones

1.2.1 Descripción de la asignatura

La Asignatura Introducción a la Programación de Arquitecturas de Altas Prestaciones está orientada a la presentación de técnicas de programación de arquitecturas hardware de altas prestaciones mediante una metodología que permite al programador obtener códigos capaces de resolver problemas de la manera más rápida y eficiente posible. Estas arquitecturas pueden ser tanto monoprocesador como paralelas con memoria compartida o paralelas con memoria distribuida.

La asignatura fue impartida por los doctores Diego Cazorla, Juan José Pardo y Enrique Arias. La asignatura está dividida en tres bloques.

En el primer bloque, impartido por el Dr. D. Diego Cazorla, se abordaron conceptos generales acerca de las arquitecturas de altas prestaciones. Además, se presentaron dos técnicas para obtener soluciones más rápidas en el procesamiento de cálculos, como son la programación orientada a bloques y la paralelización del código secuencial. En este primer bloque se pidió crear un programa en lenguaje de programación 'C' que realizara el producto de dos matrices mediante diferentes criterios de acceso a las columnas ("ijk", "jki", "kji"). Así, de esta manera, se podía evaluar la eficiencia al realizar los cálculos del producto de matrices. Y además de realizar el cálculo mediante estos criterios se pidió elaborar otro programa que realizara el mismo cálculo del producto de matrices con la particularidad de que el acceso ya no sería por el orden sino dividiendo las matrices en bloques y realizando los cálculos de manera parcial sobre los bloques, lo que mejoraba el tiempo empleado en llevar a cabo los cálculos.

En el segundo bloque, impartido por el Dr. D. Juan José Pardo, se presentó la Interfaz de Pasos de Mensajes (MPI Documents 2012) (MPI en inglés), que es una librería muy popular en la programación paralela, y la librería Basic Linear Algebra (BLAS (Basic Linear Algebra

Subprograms) 2012) , que mejora el tiempo empleado en los cálculos. En este bloque se pidió crear un programa en 'C' que empleara las librerías anteriormente descritas simulando un experimento en el que el proceso padre de MPI, proceso cero, se encargaba de inicializar y distribuir en los procesos del clúster las matrices y luego realizar el cálculo parcial de las matrices distribuidas mediante una primitiva de BLAS llamada '*cblas_dgemm*'.

Por último, el tercer bloque, impartido por el Dr. D. Enrique Arias, nos presentó '*BLACS*' (BLACS 2012) (Basic Linear Algebra Communication Subprograms en inglés) una interfaz de programación estándar para mejorar el rendimiento de operaciones en algebra lineal. En este bloque se elaboró un programa igual que en los anteriores con la particularidad que lo que se requería era crear una maya de procesos, después se distribuían las matrices en la maya de procesos creados dividiendo la matriz en bloques para, por último, crear copias y distribuir las matrices globales con la subrutina de BLACS llamada '*PDGEMR2D*'.

1.2.2 Trabajo realizado

En la asignatura Introducción a la Programación de Arquitecturas de Altas Prestaciones los trabajos que se realizaron fueron los que se detallan a continuación.

En primer lugar, se usó la Interfaz de Pasos de Mensajes (MPI) y el API Basic Lineal Algebra (BLAS) con dos matrices, A y B, recopilándose el resultado de operar con ellas en otra matriz C. En la Figura 1-1 se muestran gráficamente los resultados obtenidos al ejecutar la subrutina en el clúster, el eje vertical representa el tiempo empleado en segundos y el eje horizontal el tamaño de las matrices. De la gráfica se puede deducir que existe una relación directa entre el número de procesos destinados a realizar los cálculos y el rendimiento obtenido. Esto se debe a que con 'MPI' cada proceso realiza las tareas asignadas de manera independiente mejorando considerablemente los tiempos de respuestas hacia los usuarios.

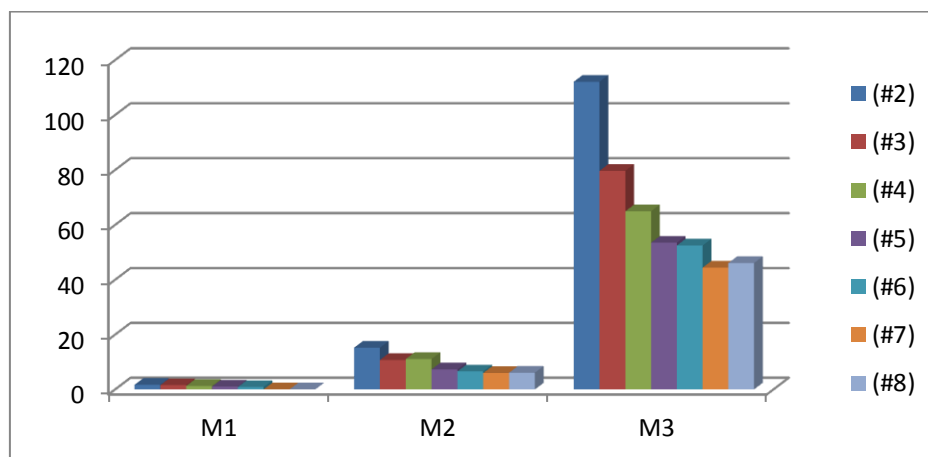


Figura 1-1. Gráfico Tiempos empleados/Matriz con MPI y BLAS

(elaboración propia)

Una segunda actividad, consistió en hacer uso de la Interfaz de Paso de Mensajes (MPI) y de Parallel Basic Linear Algebra Subprograms (PBLAS Home Page 2012) (PBLAS) con dos matrices A y B, y calcular $C=A*B$. Para lograr esto lo que se hizo fue crear un grid de proceso lógico, después distribuir las matrices en el grid de proceso y, por último, realizar la invocación a las rutinas disponibles para trabajar con sub-matrices.

Por lo que podemos observar de la Figura 1-2 si aumentamos el número de nodos y el número de procesadores por nodos, el tiempo empleado para realizar el cálculo del producto de matrices el $C=A*B$ aumenta con respecto a si solo aumentamos el número de procesadores. Por ejemplo, comparando el tiempo empleado por el proceso #4 y las matrices de tipo M3 (4000x4000), que emplea “29,498048 segundos”, con respecto al tiempo para el mismo número de procesos (#4) y tipo de matriz M3, pero con la particularidad de que el número de “Nodos” es 6, el tiempo que necesario es de “14,381442 segundos”.

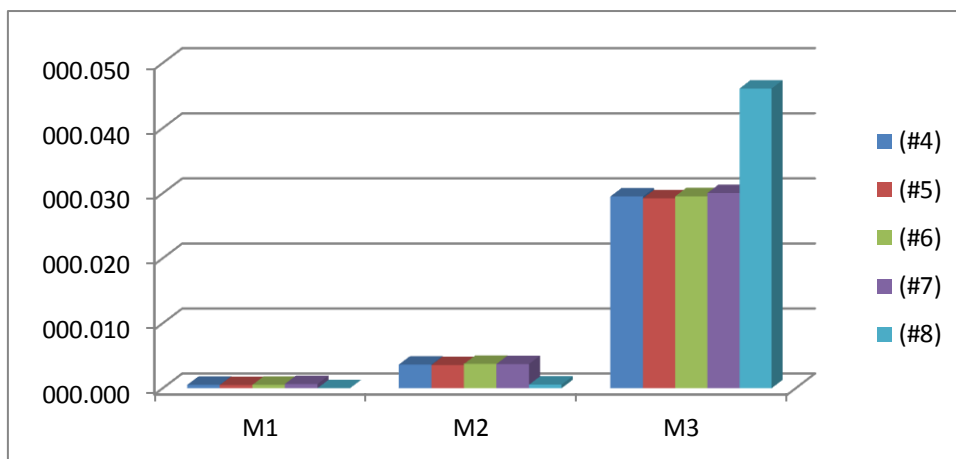


Figura 1-2. Gráfico Tiempos empleados/Matriz MPI y PBLAS

(elaboración propia)

Finalmente, relacionado con la última de las secciones y trabajos realizados, se hizo una presentación descriptiva e introductoria de una herramienta ampliamente utilizada por empresas como Yahoo! y Facebook para manejar grandes cantidades de datos de forma eficiente llamada Hadoop (Chuck 2011). La herramienta Hadoop permite el procesamiento distribuido de grandes conjuntos de datos a través de clúster de ordenadores mediante un modelo de programación simple. Esta herramienta está diseñada para manipular datos utilizando miles de máquinas, cada una de estas máquinas contribuye realizando cálculos localmente. En lugar de confiar en el hardware para ofrecer alta disponibilidad, la herramienta está diseñada para detectar y controlar los errores en la capa de aplicación, por lo que ofrece un alto nivel de servicio y disponibilidad.

Las conclusiones alcanzadas en esta asignatura son las que se recopilan a continuación:

- En la primera parte usando MPI BLAS podemos ver que si aumentamos el número de procesos se obtiene un mejor rendimiento y, por consiguiente, los resultados se devuelven con mayor rapidez.
- Usando MPI y PBLAS notamos que si aumentamos el número de procesos también el rendimiento mejora la realización de los cálculos.
- Hadoop es una herramienta versátil que permite a los usuarios hacer uso de computación distribuida.

1.3 Sistemas Inteligentes Aplicados a Internet

1.3.1 Descripción de la asignatura

La asignatura Sistemas Inteligentes Aplicados a Internet presenta el enfoque probabilístico que puede identificarse debido a las relaciones de dependencia e independencia entre conjuntos de variables ligadas a los datos y representa gráficamente, mediante grafos, estas relaciones.

La asignatura fue impartida por los doctores Dra. Julia Flores, Dr. José Miguel Puerta y Dr. Ismael García Varea. La asignatura fue dividida en tres bloques.

En el primer bloque, impartido por la Dra. Julia Flores, se introdujeron conceptos relacionados con los sistemas inteligentes, tales como probabilidades, redes bayesianas, dependencia e independencia entre variables.

En el segundo bloque, impartido por el Dr. José Miguel Puerta, se profundizó en aspectos relacionados con la construcción de redes a partir de un conjunto de datos, lo que se denomina aprendizaje paramétrico. Se presentaron modelos de estimación de parámetros como el de máxima verosimilitud o el de la estimación bayesiana, que permiten identificar las probabilidades asociadas a un conjunto de datos dado.

En el último bloque de esta asignatura, impartido por el Dr. Ismael García, se abordaron aspectos sobre Meta-heurística y minería de datos. La parte relacionada con las Meta-heurísticas consistió en aplicar un conjunto de algoritmos, tales como los algoritmos genéticos y el algoritmo de estimación de distribuciones, útiles para resolver distintos problemas. En lo que se refiere a la parte de la minería de datos se presentó una herramienta denominada WEKA (Weka 3 - Data Mining with Open Source Machine Learning Software in Java 2012), útil para modelar clasificadores probabilísticos a partir de bases de datos.

1.3.2 Trabajo Realizado

En la asignatura de Sistemas Inteligentes Aplicados a Internet se realizaron distintos ejercicios que permitieron aplicar los conocimientos adquiridos en cada bloque reseñado.

En el primer bloque se aplicaron los conocimientos adquiridos sobre estimación, enfoque bayesiano e inferencia. Además de lo dicho anteriormente, se aplicaron de manera práctica algoritmos como el de eliminación de variables en una red bayesiana.

En el segundo bloque se aplicaron técnicas de aprendizaje paramétrico, como el de estimación por máxima verosimilitud, y métricas para obtener la mejor estructura de datos. Entre las métricas utilizadas están: la métrica Bayesian Information Criterion (BIC) y la métrica BDeu (Bayesian Dirichlet y e por equivalencia de verosimilitud y u por uniforme).

En el tercer bloque se resolvieron una serie de problemas mediante la implementación de algoritmos genéticos y algoritmos de estimación de distribuciones. También se implementó una aplicación que para trabajar en el ámbito de la minería de datos, utilizando WEKA.

1.4 Calidad de Interfaces de Usuario: Desarrollo Avanzado

1.4.1 Descripción de la asignatura

La asignatura Calidad de Interfaces de Usuario: Desarrollo Avanzado está orientada a la mejora de la calidad de los sistemas interactivos, es decir, al logro de un conjunto de atributos o características que debe poseer un producto o servicio software para elevar su competitividad.

La asignatura fue impartida por los doctores Pascual González López, Víctor López-Jaquero y Francisco Montero Simarro. La asignatura está dividida en tres bloques.

En el primer bloque impartido por el Dr. D. Pascual González López nos presentó, de forma general, nociones de calidad y cómo ésta debería de ser considerada. Además, introdujo conceptos relacionados con la consideración del usuario.

Luego en el segundo bloque, impartido por el Dr. D. Víctor López-Jaquero, se presentaron nociones generales del concepto de Entorno de Desarrollo de Interfaces de Usuario basado/dirigido por modelos. Entre otras propuestas se presentó MDA (Model-driven Architecture) que es un marco de trabajo para trabajar el desarrollo de software dirigido por modelos. Es decir, el desarrollo de productos software estará basado en la especificación de diferentes vistas o modelos de ese producto software y desde estos modelos se puede llegar a generar automáticamente el código. Además, también se nos presentó el desarrollo de interfaces de usuario basado en modelos, conocido como Mb-UIDE, que son propuestas que se han presentado en el ámbito del desarrollo de interfaces de usuario y que también se apoyan en la elaboración de modelos para conseguir desarrollar la parte dedicada la interfaz de usuario de cualquier producto software interactivo.

El tercer bloque de la asignatura, impartido por el Dr. D. Francisco Montero, abordó factores que tienen que ver con la calidad en el desarrollo de interfaces de usuario. Dentro de estos elementos encontramos la calidad centrada en el uso, relacionada con la efectividad, eficiencia, seguridad y satisfacción con la que los usuarios realizan sus tareas. Otro concepto relacionado con el anterior, que también se nos ha presentado en la segunda sesión, fue el de la usabilidad, concepto anterior al de calidad en uso y muy vigente todavía en la actualidad, y que ofrece tanto dimensiones a nivel de producto como de proceso. Por último, y como instrumento para documentar la experiencia de diseño y lograr calidad en el diseño de interfaces de usuario, se han presentado los patrones de interacción, que es experiencia de diseño de utilidad en el desarrollo de interfaces de usuario. En las clases recibidas se ha sugerido la asociación de modelos a la descripción y documentación de los patrones de interacción.

1.4.2 Trabajo realizado

En la asignatura Calidad de Interfaces de Usuario Desarrollo Avanzado se presentó parte del trabajo de investigación el cual consistía en mostrar de forma general los retos y tendencias que presentan las técnicas de análisis y modelado de tareas en la actualidad a través de la definición de una serie de criterios que debería de tener una notación ideal de tal manera que cubra las necesidades actuales y futuras que presenta el desarrollo de interfaces de usuario.

A partir de la definición de una serie de criterios o características que consideramos debería de tener una notación de modelado de tareas se realizaron dos experimentos a un grupo de expertos tanto de ingeniería de software como de interacción persona ordenador. El primero de ellos consistió en asignar un grado de importancia a cada criterio para darnos cuenta el valor que representan estos para el grupo de expertos. El segundo experimento consistía en evaluar cada una de las notaciones existentes que se han propuesto en el modelado de tareas para darnos cuenta con qué grado soportan las mismas la serie de criterios que habíamos definido. Para obtener un nivel de detalle mayor (ver Capítulo 3) en donde se explica el trabajo de investigación.

1.5 Tecnología de Software Orientada a Objetos

1.5.1 Descripción de la asignatura

En la asignatura Tecnología de Software Orientada a Objetos se presentan pautas para el desarrollo de software conducido por modelos (Model-Driven Development o MDD) el cual es un concepto relativamente nuevo, con el que se intenta capturar todos los aspectos más importantes en el diseño de productos software utilizando un conjunto de modelos formales o semi formales.

La asignatura fue presentada por los doctores Dña. Elena María Martínez, Dña. María Dolores Lozano y Don. Víctor Ruiz Penichet. La asignatura está dividida en tres bloques.

En el primer bloque, impartido por la Dra. Dña. Elena Navarro, se nos presentó una introducción al desarrollo dirigido por modelos (MDD) y dentro de los muchas propuestas ligadas a MDD encontramos una propuesta, de la OMG, como es MDA, Domain Specific Modeling (DSM), y Software Factories. MDA está compuesto por distintos grupos de modelos, entre ellos: Computationally Independent Model(s) (CIM) que captura los requerimientos del sistema, The Platform-Independent Model(s) (PIM) abarca únicamente la funcionalidad y no contempla aspectos relacionados con la plataforma y The Platform-Specific Model(s), PSM, que sí que considera los aspectos dependientes de la plataforma. Además, las clases de este primer bloque tuvieron un enfoque práctico en las que se realizaron un conjunto de prácticas para implementar los conceptos descritos a través de eclipse, las cuales consistieron en crear modelos, convertir estos modelos a otros modelos y generar el código de estos modelos a través de entidades que venían representadas también en modelos.

En el segundo bloque la Dra. Dña. María Lozano abarco aspectos generales del desarrollo de interfaces dirigido por modelos (Mb-UIDE) a los cuales clasificó atendiendo a distintas generaciones en función del grado de automatización alcanzado. Posteriormente nos presentó un entorno metodológico basado en modelos para el desarrollo de interfaces de usuario, llamado IDEAS (Interface Development Environment within OASIS). IDEAS está compuesto por cuatro fases. Estas fases son las de requerimientos; dentro de esta etapa están representados los modelos de casos de uso, tareas y usuario, la de análisis; representada por diagramas de secuencia (I y II) y los diagramas de roles, la tercera etapa es el diseño; abarcando al modelo de dialogo, y, finalmente, la última etapa es la de implementación; representada por el modelo de presentación al que van ligados los objetos abstractos y concretos de interacción (AIOs y CIOs).

En el tercer bloque el Dr. D. Víctor Ruiz introdujo conceptos relacionados con los entornos colaborativos conocidos como CSCW (Computer-Supported Cooperative Work) los cuales nos permiten representar actividades de cooperación, colaboración y coordinación a través de sistemas. Posteriormente nos presentó una metodología para el desarrollo de interfaces en entornos colaborativos llamada TOUCHE (Task-Oriented and User-Centred Process Model for Developing Interfaces for Human-Computer-Human Environments) la cual está basada en la especificación de distintos modelos. Estos modelos son el modelo organizacional, el modelo de sesión, el modelo de tareas, y finalmente el modelo de objetivos.

1.5.2 Trabajo realizado

En la asignatura Tecnología de Software Orientada a Objetos se presentó una discusión sobre dos técnicas de análisis y modelado de tareas que son utilizadas con frecuencia. A partir de la definición de una serie de criterios que se presentan en el desarrollo de interfaces de usuario se realizaron dos experimentos a un grupo de expertos tanto de ingeniería de software como de interacción persona ordenador. En el primer experimento se le pidió a un grupo de expertos de ambas áreas dar un grado de importancia a los criterios definidos. Luego en el segundo experimento se procedió a evaluar cada una de las notaciones; las basadas en UML (Rumbaugh 2000) por parte del grupo de expertos de ingeniería de software y CTT (Paternò 2000) por parte del grupo de interacción persona ordenador.

La discusión se centró en valorar a través de la experiencia de un grupo de expertos el nivel de importancia y el grado con el que soportan una serie de criterios que debería de tener una técnica/notación de modelado de tareas.

Las técnicas basadas en UML resultaron ser más apropiadas desde el punto de vista global ya que se alcanzó una valoración más alta (-38) que con respecto a CTT (-75). Sin embargo la conclusión a la que se llegó fue que ninguna de las notaciones cubren con satisfacción los criterios propuestos tales como el de aplicaciones software altamente interactivos o la escalabilidad que estas pudieran tener.

1.6 Programación Internet con Lenguajes Declarativos Multiparadigma

1.6.1 Descripción de la asignatura

La asignatura Programación internet con lenguajes declarativos multiparadigma está orientada a presentar conceptos relacionados con la búsqueda en Internet a través del uso de lenguajes declarativos que integran la programación lógica y la lógica difusa (fuzzy logic).

La asignatura fue impartida por los doctores Pascual Julián Iranzo, Francisco Romero Chicharro y Ginés Moreno Valverde. La asignatura está dividida en tres bloques.

En el primer bloque, presentado por el Dr. D. Pascual Julián Iranzo, se hizo una breve introducción a los lenguajes imperativos más comunes hoy en día, tal como 'C', y los lenguajes declarativos, que se diferencian de los anteriores en que los programas se definen como un conjunto de fórmulas lógicas y los primeros como referencias a variables en memoria, además introduce fundamentos de la programación lógica, basada en fragmentos de lógica de predicados, abordando los conceptos de unificación y resolución. Luego se presentó la

semántica operacional (mecanismo de computo que permiten ejecutar los programas) y declarativa en la programación lógica y esta misma programación basada en la resolución débil.

En el segundo bloque, presentado por el Dr. D. Francisco Romero, se nos presentó una amplia variedad de aplicaciones que conjugan programación declarativa e Internet. Además también se describió cómo ha ido evolucionando la Web desde su versión 1.0, en donde el contenido presente se caracterizaba por su carácter estático hasta lo que hoy se está desarrollando y se conoce como web 2.0 o bien Web semántica. Otro punto abordado en este bloque es el de la búsqueda en la Web que se da a través de tres tipos de mecanismos: los directorios web, los buscadores y los meta-buscadores (hacen uso de otros buscadores para obtener resultados). También se profundizó en las limitaciones que encuentran hoy en día los buscadores porque la información textual de los sitios web no está estructurada y los algoritmos en que estos se basan únicamente en la aparición de palabras.

Y por último, el tercer bloque impartido por el Dr. D. Ginés Moreno hizo una descripción breve de Prolog como lenguaje natural de programación lógica. Además nos presentó FLOPER, herramienta utilizada para realizar búsquedas del tipo lógico difusas basadas en la extensión del lenguaje MALP (Vazquez and G 2011) que han desarrollado en su grupo de investigación. Luego nos presentó Fuzzy XPATH (Luna and Moreno 2012) el cual es una extensión del lenguaje de búsquedas a través de documentos XML llamado XPATH con la particularidad que en esta extensión se introducen aspectos difusos como la media aritmética de la probabilidad asociada a conceptos diferentes.

1.6.2 Trabajo realizado.

En la asignatura de Programación Internet con Lenguajes Multiparadigmas se presentó como trabajo de investigación la relación existente entre el desarrollo de interfaces de usuario y Prolog. En ella se pudo abordar que existen varias soluciones para hacer que los sistemas orientados a objetos estén disponibles en Prolog.

Una de estas soluciones trata de una rígida separación de la interfaz gráfica de usuario, desarrollado en un entorno de desarrollo GUI externo, de la aplicación. Un estrecho puente que enlaza el sistema externo a Prolog, y el desarrollo de GUI a través de una extensión de Prolog. Y el otro enfoque general es tratar de extender Prolog para que permita agregar una serie de controles o elementos de GUI. Algunos estilos en los que se ha tratado de aplicar ambos conceptos para trabajar con ellos son tales como el desarrollo de GUI externo en los que se usa un lenguaje externo como base para crear las interfaces gráficas y se le añade el motor de Prolog de forma externa a través de protocolos de comunicación local o en red. Otras de las formas de relacionar GUI y Prolog es a través de la inserción de código Prolog en lenguajes de programación externo. Por último la forma menos común de relacionar Prolog y GUI es a través de la extensión de la API existente para que puedan manejar predicados en Prolog.

Capítulo 2

Estado del Arte

2.1 Introducción, contexto y alcance de la investigación: Nuevos paradigmas de interacción

Un paradigma es un ejemplo o modelo de los que se derivan todos los sistemas de interacción, es decir un paradigma es una abstracción de todos los posibles modelos de interacción organizados en grupos con características similares. Así que, a partir de esta definición podemos decir que entendemos por nuevos paradigmas de interacción, entre el usuario y la máquina, los modelos de los que se derivan todos los sistemas de interacción entre los usuarios de las aplicaciones software y las máquinas (ordenadores, teléfonos inteligentes, tabletas, etc.). Pero, ¿qué se entiende por interacción entre persona y máquina? Veamos a continuación cómo se define este concepto.

La interacción es un proceso que se da a través de la comunicación entre dos o más objetos, estos objetos pueden ser los usuarios (seres humanos) de las aplicaciones y las aplicaciones o sistemas (máquinas). La interacción se realiza a través de lo que se conoce como interfaz, que es el medio que permite el proceso de comunicación entre ambos (usuario y sistema) con el fin de que la interacción tenga éxito. Estos conceptos son los que definen la Interacción Persona Ordenador (IPO) también conocida como Human Computer Interaction (HCI en inglés) la cual es una disciplina que abarca aspectos como el diseño, la evaluación y la implementación de aplicaciones software orientadas a las personas.

La IPO abarca no solo aspectos relacionados con los seres humanos y las máquinas, de hecho es un área interdisciplinar ya que se tienen en cuenta áreas que no están necesariamente relacionados con las computadoras y la interacción que las personas pudiesen tener con estas, tales áreas van desde la psicología (relacionado a los procesos cognitivos), la sociología (la interacción entre las personas) pasando por el diseño industrial (específicamente lo que tiene que ver con los productos).

Ya que es un área interdisciplinar, la IPO ha ido sufriendo cambios debido a la necesidad de adaptarse a la propia evolución tecnológica. Estos campos han sido las computadoras gráficas, los sistemas operativos, factores humanos, ergonómicos, la ingeniería industrial y los sistemas como partes de las máquinas. El uso de dispositivos de almacenamiento en los ordenadores gráficos fue lo que permitió el desarrollo de técnicas de IPO. Por otro lado, en el ámbito de los sistemas operativos se desarrollaron técnicas que permitieron la conexión de dispositivos de entrada y salida dando lugar a tiempos de respuestas ligados a la interacción humana lo que permitió el desarrollo de sistemas de gestión y herramientas de interfaces de usuarios. Los factores humanos están relacionados a la necesidad de diseñar equipos que puedan ser utilizados por las personas y la ergonomía se diferencia de los factores humanos en que se enfoca más al estudio de cómo y con qué trabaja el usuario. En la ingeniería industrial se prioriza el diseño eficiente de manuales para trabajar a través de herramientas especializadas

para incrementar la productividad y reducir el riesgo de fallo. Y por último el auge que ha tenido la informática vinculados a la calidad de las interfaces de usuario.

En el pasado cuando se hablaba de interacción entre las personas y las maquinas se tenía en mente a una persona que usaba un programa grafico interactivo en una estación de trabajo. Sin embargo el concepto de máquina no está limitado a una estación de trabajo sino que se amplía a una gran variedad de dispositivos que podrían ser diseñados a partir de que se dé una relación explícita entre las máquinas y los seres humanos, tales dispositivos podrían ser por ejemplo un horno de microondas cuyo diseño está pensado específicamente en serle intuitivo a las personas que hacen uso de ello.

La Figura 2-1 presenta los cinco principales aspectos, identificados por la ACM¹, que tienen que ver con el carácter interdisciplinar de la interacción persona-máquina. Estos aspectos son los siguientes: (N) la naturaleza de la interacción persona-máquina, (U) el uso y el contexto de las máquinas, (H) las características humanas, (C) sistemas máquinas y la arquitectura de la interfaz, y (D) el proceso de desarrollo.

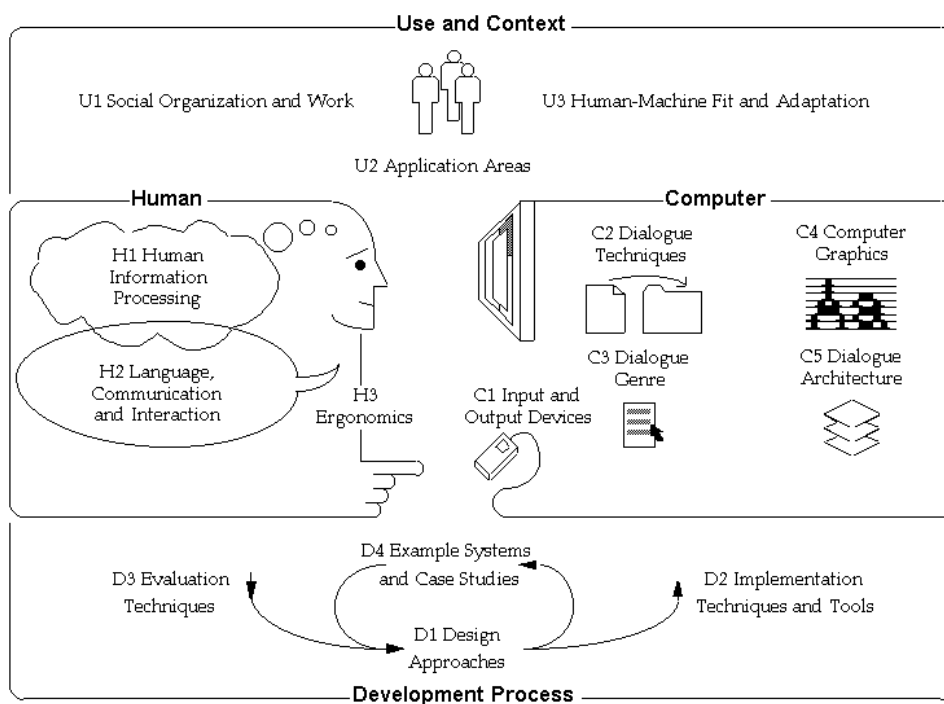


Figura 2-1. Alcance y dimensión de la disciplina Interacción Persona-Ordenador

La propuesta de Trabajo Fin de Máster que se presentará en este documento abarca dos aspectos relacionados con la interacción. Estos factores están incluidos dentro de la sección dedicada al (D) proceso de desarrollo. Dentro de este apartado la aportación que propone esta propuesta de tesis doctoral estará incluida dentro de las actividades destinadas a considerar los aportes de diseño que incluyen técnicas de análisis de tareas, las técnicas de especificación de diseño y análisis y las técnicas y herramientas de implementación. Es decir, nuestra

¹ <http://old.sigchi.org/cdg/cdg2.html>

propuesta está relacionada con las estrategias y herramientas de implementación y con las relaciones entre el diseño, la evaluación y la implementación.

En este capítulo se dará una visión general de los nuevos enfoques de Interacción Persona-Ordenador con el fin de sentar las bases teóricas de los aspectos más relevantes que tienen que ver con el desarrollo de interfaces de usuario en diferentes dominios.

Este capítulo estará organizado como sigue. En primer lugar, se identificarán los distintos estilos de interacción que pueden considerarse al soportar la interacción hombre-máquina. Después identificaremos las tendencias que se utilizan actualmente para abordar el desarrollo de interfaces de usuario. Entre esas tendencias destacan los desarrollos de interfaces de usuario basados en modelos (Mb-UIDE). La principal característica de esa propuesta es la utilización de distintos modelos. Uno de esos modelos es el dedicado a las tareas. Por ello, identificaremos y analizaremos las distintas técnicas y notaciones disponibles para dar soporte a las actividades de análisis y modelado de tareas. Finalizarán este capítulo dos apartados. Uno de ellos estará dedicado a la identificación de trabajos, revistas, conferencias y grupos de investigación especialmente relacionados con la difusión y la investigación en el ámbito del análisis y el modelado de tareas, y un último apartado dedicado a recopilar las principales conclusiones alcanzadas con la realización de este capítulo.

2.2 Interacción hombre máquina

En esta sección se abordarán aspectos relacionados con los estilos de interacción que se conocen actualmente, entendiéndose como estilo de interacción la forma en que los usuarios se comunican con las aplicaciones software, y los paradigmas de interacción que se han venido desarrollando. Los contenidos recogidos en esta sección están justificados desde el punto de vista de que se hace necesaria una identificación inicial de la diversidad en los estilos y paradigmas de interacción, que puede utilizar un usuario cuando interactúa con una máquina.

2.2.1 Estilos de interacción hombre-máquina

El término estilo de interacción² se utiliza frecuentemente para describir la forma en que los usuarios se comunican o interactúan con un ordenador. Estas formas de comunicarse o interactuar por parte de las personas varían constantemente en función de la naturaleza de la interacción y de cómo estas han ido evolucionando a lo largo del desarrollo en el campo de las ciencias de la computación.

El primer estilo de interacción conocido en el campo de la computación fue el denominado interfaz por línea de comandos el cual tiene sus orígenes y popularización en el desarrollo de ciertos sistemas operativos como UNIX y posteriormente LINUX o MS-DOS. Hoy en día este estilo de interacción aún se sigue utilizando con frecuencia, ya que muchas personas que interactúan a través de una terminal se sienten satisfechas porque les da la sensación de que controlan todo en lo relacionado con sus aplicaciones software. La idea básica en este estilo de interacción es tratar de dar instrucciones directamente al ordenador a través de la entrada de una secuencia de caracteres. Estas instrucciones pueden tener la forma de las teclas de

² <http://www.slideshare.net/ximenatabares/paradigmas-de-interaccion-usuariomquina>

función (por ejemplo la tecla F1), pueden ser un solo carácter, abreviaciones cortas, palabras enteras o una combinación de las primeras. La Figura 2-2 muestra un ejemplo de una línea de comandos. Sin embargo este estilo de interacción presenta ciertas características que lo hacen un poco difícil de manipular para aquellas personas que no están familiarizadas con lenguajes de programación o líneas de código ya que es un poco tedioso tener que aprenderse la nomenclatura con un lenguaje específico.

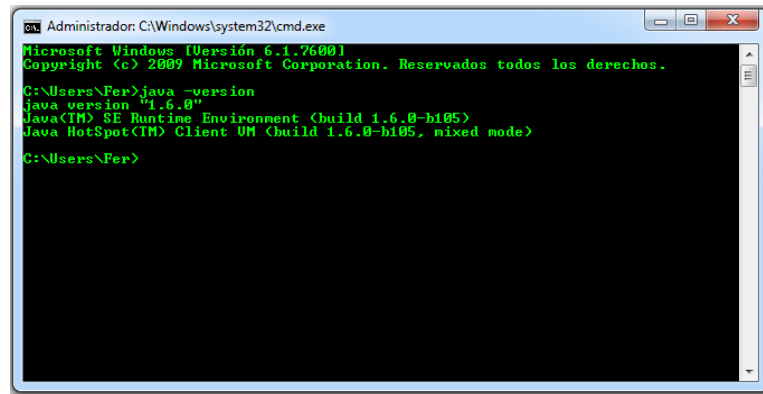


Figura 2-2. Grafica de interfaz por línea de órdenes

Debido a que el proceso en el que las personas interactuaban a través de líneas de ordenadores era tedioso surge la necesidad de crear otros estilos de interacción que fuesen más intuitivos para las personas que se veían inmersas en realizar ciertas actividades. Es así como nace otro estilo de interacción que evita tener que retener y aprender una gran cantidad de comandos denominado 'Form fillin' o relleno a través de formularios. En este estilo de interacción se trata de guiar a los usuarios a través de reglas predefinidas mediante la complementación de espacios vacíos o campos de información mediante un formulario para luego ser almacenados en algún tipo de sistemas de base de datos (archivos de texto por ejemplo). La Figura 2-3 muestra un ejemplo del estilo de interacción basado en formularios en el cual se intenta realizar el inicio de sesión de un usuario ingresando cierta información que solo él conoce.



Figura 2-3 Ejemplo de interacción basada en cumplimentación de formularios

Además del estilo de interacción mencionado en el párrafo anterior existe otro estilo que trata que los usuarios puedan seleccionar entre un conjunto de alternativas denominado Menú. Un menú es un conjunto de opciones a ser visualizadas en una pantalla para brindarle al usuario la oportunidad de elegir entre ellas. Los usuarios pueden elegir entre una o más y esa elección genera eventos, cuyo tratamiento supone la ejecución de una orden y por lo general provoca

un cambio en la interfaz. El estilo de menús se diferencia del estilo de la interfaz por línea de órdenes en que los usuarios tienen la ventaja de no tener que recordar palabras o sintaxis siempre y cuando el texto presentado en el menú clarifique la acción que el usuario necesita ejecutar. La Figura 2-4 presenta un ejemplo de menú de la herramienta denominada IdealXML desarrollada en (F. Montero 2005).

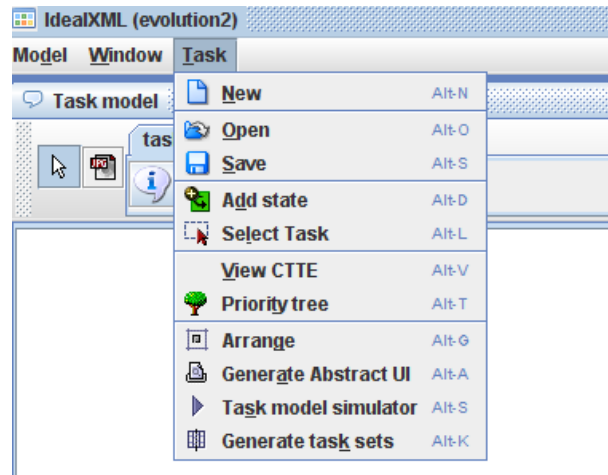


Figura 2-4. Ejemplo de interacción basada en menús

Un estilo de interacción que ocupa una parte distinguida en el desarrollo de interfaces de usuarios ya que es muy usado hoy en día en muchas aplicaciones es la manipulación directa. Este estilo de interacción está basado en el enfoque de ‘arrastrar y soltar’, es decir manipular objetos a través de acciones y estas acciones a su vez pueden provocar cambios o efectos que se pueden ver en los objetos que se manipulan. La Figura 2-5 muestra un ejemplo de este estilo de interacción.

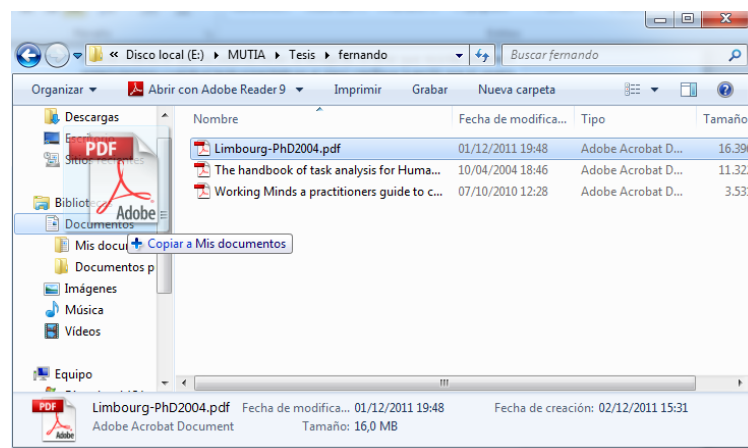


Figura 2-5 Ejemplo de interacción basada en la manipulación directa

Y por último, pero no menos importante, está el estilo de interacción asistida el cual se refiere al uso de un asistente personal o un agente que coopera con el usuario en la realización de una tarea y los usuarios en vez de dirigir la interacción trabaja en un entorno cooperativo en el que los usuarios y asistente/agentes se comunican entre sí controlando eventos y realizando tareas. Un agente de la interfaz es un programa que puede ser considerado por el usuario como un asistente que le ayuda al usuario a realizar determinada tarea.

Una vez que hemos tratado con los estilos de interacción, a continuación veremos enfoques o paradigmas asociados a la interacción entre personas y máquinas

2.2.2 Paradigmas y modalidades de interacción.

Los paradigmas de interacción actuales, como ya hemos dicho, se clasifican en función de la forma en la que el usuario interactúa con el sistema. Así, de esta manera encontramos que una de las primeras formas de interactuar de las personas con las máquinas fue a través de un ordenador denominándose a este enfoque como paradigma del ordenador personal. En el paradigma del ordenador personal la interacción se realiza explícitamente entre un usuario y el ordenador limitando la interacción a este entorno siendo ésta una de las principales características que encontramos. Posteriormente, con el desarrollo de las ciencias de la computación, surgieron otras formas de interactuar entre las personas y las máquinas. De esta manera surgió un enfoque en el que los usuarios (denominados avatares) creaban su propio mundo y lo manipularan a su antojo, este paradigma se denominó realidad virtual. La realidad virtual es un enfoque en el que se realiza una simulación en los ordenadores a través de gráficas en el que el usuario pueda crear un mundo virtual y controlar lo que en él sucede.

Posteriormente, debido a la necesidad de realizar tareas a través de la colaboración de múltiples usuarios nace la computación ubicua en la que existen usuarios conectados en una red de ordenadores. En el enfoque de la computación ubicua, muchas veces asociado a los entornos colaborativos, la idea básica reside en que si se requieren realizar un conjunto de actividades o tareas se hagan a través de la colaboración entre varios usuarios, simulando el proceso que se lleva a cabo en el mundo real. Un ejemplo de esto sería simular, en un mundo virtual, el levantamiento de una viga de madera en el que en cada extremo de la viga estaría actuando un usuario y el efecto que uno realice sea percibido por el usuario adyacente en el otro extremo a través de un conjunto de elementos definidos por el diseñador dando lugar a la colaboración de las actividades que se llevan a cabo en este proceso.

Por último, surge la opción de tomar ideas que se crean en un mundo virtual y trasladarlas al mundo real y conocido, generando lo que se conoce como realidad aumentada. La realidad aumentada intenta retomar abstracciones que los usuarios crean en la realidad virtual. Por ejemplo, nos podríamos encontrar con la idea de que la forma de interaccionar a través de dispositivos sensibles al tacto como los teléfonos móviles se pudiesen haber creado en un mundo virtual y luego esa idea haya sido trasladada al mundo real.

Se debe notar en este punto que se tiene que hacer una diferenciación entre el primer paradigma (el ordenador personal) y los demás (nuevos paradigmas) ya que a estos últimos se les denomina paradigmas de interacción post-WIMP debido a que anteriormente la interacción se daba explícitamente a través de los usuarios y los ordenadores personales mediante mecanismos tales como Ventanas (Window), iconos (Icons), etc., conociéndose a esta forma de interactuar como WIMP, pero ahora la interacción no está limitada a un ordenador personal sino que existen una gran cantidad de dispositivos como gafas 3D, tabletas, guantes, reconocimiento gestual a través de ordenadores, etc. .

A partir de esos enfoques se han ido desarrollando diferentes modalidades en las que los seres humanos podrían interactuar con las máquinas; estas modalidades se han clasificado en tres tipos comúnmente reconocidas hoy en día como las interfaces gráficas de usuario (Graphical

User Interface o simplemente GUI en inglés), la interfaz de usuario por voz (Voice User Interface o VUIs), y las interfaces multi-modales (Multi-Modal Interface).

Una vez que hemos identificado el alcance y diversidad asociada a la interacción y a las interfaces de usuario veremos cómo se puede atacar esta diversidad y qué métodos, técnicas y entornos se están utilizando para el desarrollo de interfaces de usuario.

2.3 Desarrollo de interfaces de usuario

La técnica más ampliamente extendida y utilizada para abordar el desarrollo de interfaces de usuario es la conocida como Entorno de Desarrollo de Interfaces de Usuario Basado en Modelos (Mb-UIDE). El desarrollo de interfaces de usuario basado en modelos tiene su origen en la década de los noventa (Schlungbaum 1996); (Puerta 1997); (Limbourg 2003)). Bajo estos entornos se propone un conjunto de abstracciones (modelos), un proceso de desarrollo y unas herramientas para soportar el desarrollo de interfaces de usuario. La Figura 2-6 muestra los componentes típicos y los principales procesos de desarrollo en Mb-UIDE.

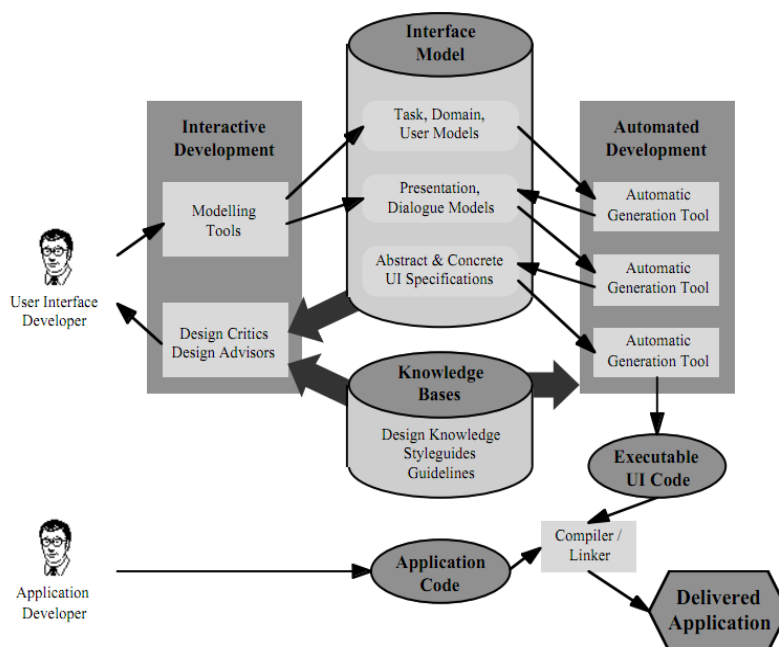


Figura 2-6. Esquema general asociado a un entorno de desarrollo de interfaces de usuario basado en modelos

(fuente: Schlunbaum, 1996)

La idea principal que subyace bajo los Mb-UIDE según (F. Montero 2005) es la elaboración de modelos conceptuales en los que se recogen características tanto estáticas como dinámicas de una aplicación, tales características pasan por ser partes de la interfaz de usuario, las tareas que se desean realizar, los datos que se desean manipular y las transformaciones que ligan toda esta información.

Seguidamente identificamos algunos de los principales modelos, junto a sus aportaciones, que se han identificado como necesarios en las técnicas de desarrollo de interfaces de usuarios basado o conducido por modelos.

Uno de los primeros aportes en Mb-UIDE fue el propuesto por (Puerta 1997) el cual definía un conjunto de modelos que se caracterizaban por el ámbito en el cual se aplicaban, así de esta manera se presentaban los modelos de dominio, usuario, dialogo y tareas. El modelo de dominio captura conceptos semánticos del dominio de la aplicación. El modelo de dominio (Lopez-Jaquero 2005) incluye una visión de los objetos sobre los que actúan las tareas capturadas. El modelo de usuario (Limbourg 2003) propone abstracciones para mejorar la comprensión, el razonamiento y la manipulación de lo que es una interfaz de usuario. En un modelo de usuario o de interfaz de usuario debe capturar todas las características y requisitos de los usuarios tanto individualmente como colectivamente. El modelo de dialogo representa por lo general secuencias de entradas y salidas. Y finalmente, el modelo de presentación es la forma de describir la apariencia de las interfaces de usuario que serán utilizadas por los usuarios finales. La Figura 2-7 presenta las relaciones entre los modelos asociados a los entornos Mb-UIDE que hemos descrito y el nivel; semántico, sintáctico o léxico, al que pertenece cada una de estas aportaciones. Al considerarse en este trabajo los modelos de tareas nuestra aportación se realizará en el contexto del nivel semántico que es el que contiene el aporte de los modelos de tareas.

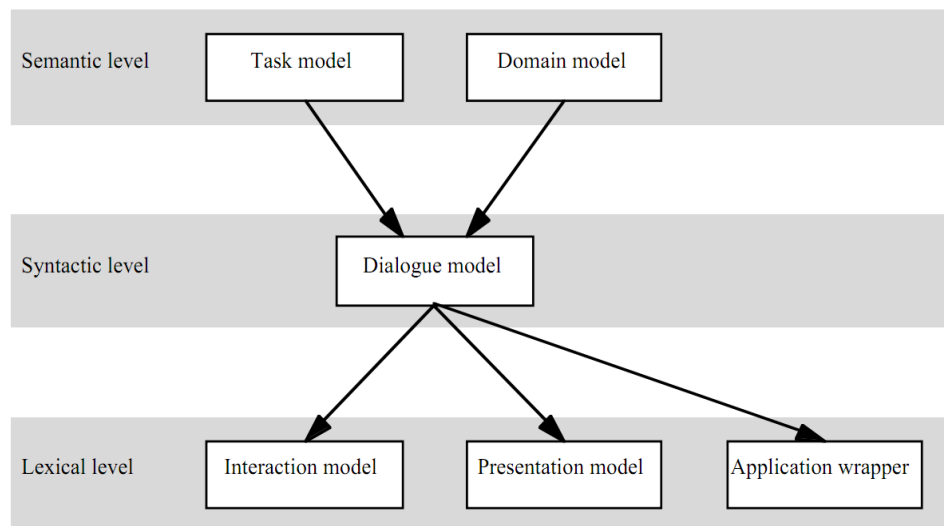


Figura 2-7. Modelos y relaciones entre ellos definidos en un Mb-UIDE

(fuente: Schlunbaum, 1996)

No obstante actualmente se ha consensuado una nueva propuesta en un proyecto europeo llamado CAMELEON (Vanderdonckt J. 2002) (Context Aware Modelling for Enabling and Leveraging Effective interaction, FP5-2000-IST2) liderado por algunos autores con notables aportaciones en Mb-UIDE. En él se redefinen los modelos y las relaciones que éstos guardan. El punto fuerte de esta propuesta está en la búsqueda de una convergencia con propuestas de desarrollo más próximas a la Ingeniería del Software y el hecho de que los modelos cobran, si cabe, un papel más relevante que en los desarrollos Mb-UIDE tradicionales.

El consenso alcanzado en el proyecto CAMELEON consiste en la identificación de dos tipos de modelos: aquellos independientes de la plataforma y aquellos dependientes de ella. En el primer nivel se localizan los modelos de dominio y tareas y en el segundo grupo están los

niveles de presentación. En este último grupo de niveles se identifican tres niveles de abstracción: nivel abstracto, concreto y final.

Aparte de estos modelos, que están muy relacionados con los modelos disponibles en las propuestas Mb-UIDE tradicionales, se identifican otros modelos, muchos de estos modelos se agrupan dentro del concepto de modelo ontológico. Los modelos ontológicos son instanciados en modelos arquetipos u observados. Los modelos arquetípicos son un conjunto de modelos declarativos que sirven como entradas de cualquier sistema y los modelos observados son modelos que soportan la adaptación durante el proceso de ejecución.

La fase de diseño en CAMELEON es básicamente un proceso de desarrollo que al final resulta en un conjunto de interfaces de usuario ejecutables en un contexto determinado. Y finalmente, la fase de ejecución consiste en construir la infraestructura y las interfaces de usuario producidas en la fase de diseño. La Figura 2-8 muestra el modelo descrito anteriormente por la propuesta de estandarización basada en modelos definida en CAMELEON.

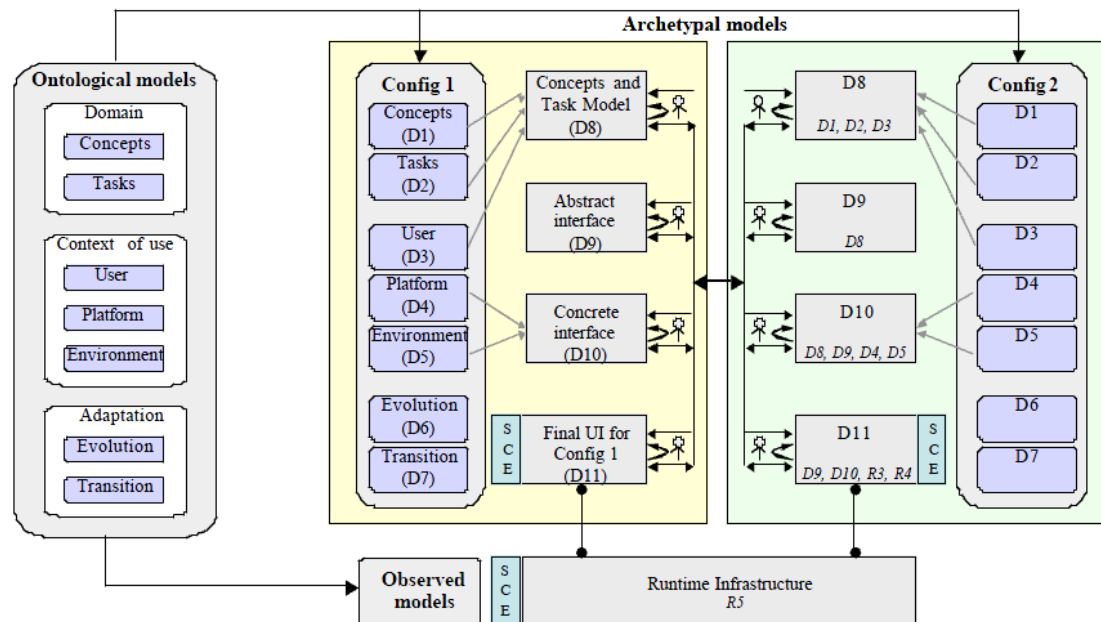


Figura 2-8. CAMELEON framework

Las relaciones entre modelos identificadas en CAMELEON conectan los distintos modelos. En este sentido, los modelos de conceptos y tareas son el origen de la especificación abstracta de la interfaz de usuario (AUI) y ésta, a su vez, permite obtener una interfaz de usuario concreta (CUI). La interfaz de usuario concreta podrá ser llevada a código final (FUI).

En el primer nivel de modelo de conceptos y tareas reúne conceptos y descripciones de tareas producidas por los diseñadores para un sistema interactivo y un contexto de uso en particular. Luego las interfaces de usuarios abstractas (AUI) son representaciones abstractas de conceptos del dominio y funciones de tal manera que es independiente del tipo de interacción o de la plataforma/dispositivo en el que se realizara la interacción. Las especificaciones abstractas AUI son convertidas en objetos de interacción concreta (CIOs) dependientes del tipo de

interacción, y es aquí donde se tiene en cuenta la plataforma en la que se ejecutara la aplicación. Finalmente, las interfaces de usuario final (FUI) generadas desde los CIOs pueden llevarse a cualquier código, como por ejemplo Java, HTML, PHP, etc.

La Figura 2-8 muestra el meta modelo de arquetipos los niveles que hemos descrito en la propuesta de Mb-UIDE que actualmente se está llevando a cabo. En este trabajo de investigación centraremos nuestra atención en el modelado de tareas y en las notaciones que dan soporte a dicha actividad.

El modelo de tareas es un modelo ampliamente utilizado en el desarrollo de interfaces de usuario. A través de este modelo se expresan explícitamente cuáles serán las tareas que los usuarios llevaran a cabo a través de un conjunto de interfaces de usuario desarrolladas. Los modelos de tareas son llevados a cabo de tal manera que su comportamiento simule de la forma más precisa el comportamiento de los usuarios en sus actividades. La especificación de los modelos de tareas es independiente de la plataforma y de la modalidad. Este tipo de modelos serán especialmente considerados en el Trabajo Final de Máster y en la propuesta de tesis doctoral que acompaña a este documento.

En la próxima sección se identificarán y describirán las principales propuestas asociadas a las técnicas de análisis y modelado de tareas. Dichas técnicas son las utilizadas para desarrollar los modelos de tareas utilizados en las propuestas Mb-UIDE.

2.4 Análisis y modelado de tareas

El análisis de tareas es uno de los primeros aportes para la obtención y el análisis de la información (Paternò 2000). El análisis de tareas se basa en las observaciones de las actividades de los usuarios consistiendo básicamente en analizar y describir las tareas que éstos realizan. Según (Diaper D. 2004) el análisis de tareas es potencialmente el método más útil para los que trabajan en la disciplina interacción persona-máquina y tiene aplicaciones en todas las etapas del desarrollo de un sistema.

Además del análisis de tareas para llevar a cabo la obtención y el análisis de la información, existe otro aspecto muy importante que también va ligado a la interacción persona máquina, denominado modelado de tareas. Esta otra actividad es el siguiente paso a realizar una vez se han llevado a cabo las actividades de análisis de tareas. El modelado de tareas, según (Lopez-Jaquero 2005), identifica y describe cuáles son las tareas que va a realizar el usuario de una aplicación mediante el uso de su interfaz de usuario. Las tareas se descomponen en acciones atómicas que representan los pasos necesarios para alcanzar los objetivos de la misma. Dentro de los datos capturados en este modelo también se recogen los requisitos no funcionales de las tareas, como son por ejemplo los requisitos de tiempo de respuesta. La definición anterior concuerda con (Limbourg 2003) en el sentido en el que define a un modelo como una simplificación, ya que retira los detalles de objetos del mundo real y trata de identificar propiedades de interés de objetos del mundo real.

En esta sección se abordaran algunas de las técnicas de análisis y modelado de tareas que actualmente se utilizan en los paradigmas de interacción persona-máquina. En la sección 2.4.1 se trataran de dar definiciones generales de los elementos que intervienen en el análisis y el modelado de tareas tales como la interacción, las metas, sub-metas, operadores, entre otros.

Posteriormente en la sección 2.4.3 se describirá el análisis jerárquico de tareas conocido como HTA el cual es un método de análisis de tareas usados en el paradigma de interacción de entornos virtuales colaborativos (CVEs en inglés) por (García 2010). Luego en la sección 2.4.4 se hablara de una técnica basada en el análisis denominada GOMS. Y por último en la sección 2.4.5 se tratara ampliamente la notación CTT que es una técnica basada en una notación llamada LOTOS.

Aunque las técnicas de análisis y modelado de tareas se utilizan para ciertos paradigmas de interacción , ejemplo el descrito en el párrafo anterior en CVEs, no está justificada de forma objetiva la utilización de un tipo de técnica de análisis y modelado de tareas en específica, más bien queda sujeto al criterio del que desarrolla su aportación en el ámbito del paradigma seleccionado por lo que conoce sobre una técnica específica, tal vez la que el crea que mejor se ajuste a lo que el supone lo que será el comportamiento de las actividades de los usuarios.

2.4.1 Definiciones previas relacionadas con modelado de tareas

En este punto se trata de dar una noción básica de los elementos que tienen que ver con el análisis y modelado de tareas y los paradigmas de interacción hombre-máquina, tales elementos son comunes entre todos los modelos, así de esta manera a continuación los describiremos.

Tareas (Tasks)

El término tarea por lo general se emplea para designar una actividad o un trabajo que demanda de parte de quien la lleva a cabo cierto esfuerzo y que se realizará durante un tiempo limitado, es decir, existe un tiempo límite para su realización. De esta manera a partir de este concepto podemos definir como tarea a una actividad que se debe realizar para llevar a cabo un proceso. Además, las tareas pueden ser definidas en términos de su(s) meta(s).

Metas (Goals) y Sub-Metas (SubGoal)

Una meta (goal) es el mejor indicador de un estado específico de las actividades, formalmente un estado meta (goal state). El estado meta (Diaper D. 2004) puede ser un evento o algún otro valor físico observable de una o más variables que actúan como criterios de la consecución de metas y, finalmente, el rendimiento del sistema. En un momento dado, una meta puede ser activa o latente. Las metas activas son los que están siendo perseguidas en la actualidad, mientras que las metas latentes son aquellas que pueden llevarse a cabo en condiciones que puedan surgir.

Las sub-metas son en esencia las mismas metas con la particularidad que estas, las sub-metas, se encuentra a un nivel jerárquico menor que el de la meta padre la que está a un nivel superior.

Operaciones (Operations) y Sub-Operaciones (Sub-Operations)

Una operación (operation) es la unidad fundamental de análisis. La operación se define por su meta (s) (o metas). Además, se especifica actividades, acciones (Action), que contribuyen a la consecución de metas, y (las condiciones indicando la consecución de metas, retroalimentación (Feedback), por lo que las operaciones se refieren a veces como unidades I-A-F.

De la misma manera en que las metas pueden descomponerse en sus respectivas sub-metas, las operaciones se pueden descomponer en sub-operaciones dispuestas en una jerarquía anidada. Las sub-operaciones se incluyen dentro de un orden de operaciones superior (o de orden superior), el logro de cada sub-meta es hacer una contribución única para lograr la meta de orden superior. La sub-operación comprende una operación de orden superior que deben ser mutuamente excluyentes y en conjunto comprenden una exposición exhaustiva de las metas parciales y metas de orden superior.

Acción (Action)

Una acción (*Action*) puede ser entendida como una medida y/o instrucción para hacer algo en determinadas circunstancias. Una acción puede ser entendida formalmente como una regla de transformación es decir, una especificación de cómo se responde a una señal de error y su cancelación.

Planes (Plans)

La especificación de la regla o reglas que rigen el orden en las sub-operaciones que debe llevarse a cabo se llama un plan. Los planes pueden ser de varios tipos, los más comunes son simplemente una secuencia fija o un procedimiento de rutina, tales como "hacer esto, entonces, entonces esto ", y así sucesivamente. Otro tipo común de plan especifica es una regla selectiva o de decisión: "Si 'x' es el caso, hacer lo siguiente. Si 'y' es el caso, hacer esto otro. "Estos dos tipos de planes son importantes porque implica el conocimiento por parte del operador. El conocimiento necesario puede ser simple conocimiento de los procedimientos declarativos o un amplio conocimiento del ambiente, los límites y capacidades de la máquina, las normas de seguridad, y muchas otras cosas.

Un tercer tipo distinto del plan requiere de dos o más operaciones que deben realizarse en paralelo. Dicho de otro modo, la meta de orden superior no puede ser alcanzada a menos que dos o más metas subordinadas se obtienen al mismo tiempo. Dicho plan se conoce como tiempo compartido (time-sharing) o plan de tareas dual (dual task plan), y este tipo de plan también tiene importantes implicaciones cognitivas de la división de la atención y en el caso de las operaciones de equipo la distribución de información entre los miembros del equipo actúan juntos.

2.4.2 Técnicas de análisis y modelado de tareas basadas en UML

Una de las propuestas ampliamente extendidas en el modelado de sistemas ha sido el lenguaje unificado de modelado (Rumbaugh 2000) (UML o Unified Modeling Language en inglés) que es un lenguaje grafico que permite visualizar, especificar y construir sistemas. Este método de modelado de sistemas cuenta con una gran variedad de diagramas que permiten representar diferentes aspectos de las entidades representadas. Dentro de estos diagramas que nos permiten representar a las entidades a través de tareas se encuentra los diagramas de casos de usos, actividades, secuencia, comunicación.

Los casos de uso describen con precisión los sistemas desde el punto de vista funcional ya que se centran en las actividades que tienen que ver con la lógica de la aplicación, su funcionamiento y posibles excepciones que esta pudiera generar, sin embargo uno de los aspectos más criticados de estos es que no permiten obtener requisitos que no estén relacionados con los procesos del sistema tales como la usabilidad que pudiera tener este.

Por otra parte, otra de las técnicas de tareas basadas en UML son los diagramas de actividades que son representaciones gráficas del flujo de las actividades de la lógica de los sistemas, en general los diagramas de actividades muestran un flujo de control general de las tareas.

La Figura 2-9 muestra un ejemplo de un diagrama de actividades de un sistema de control de crédito.

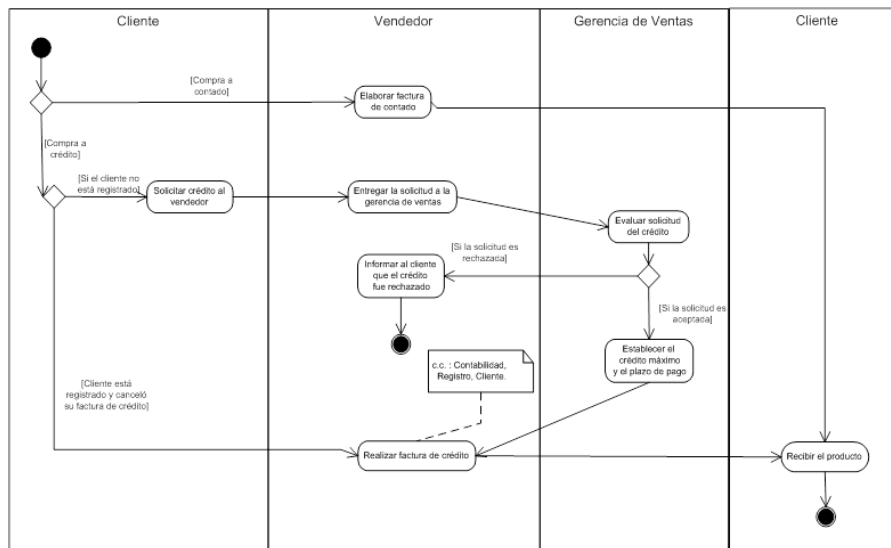


Figura 2-9 Diagrama de actividades de control de crédito

Otra técnica son los denominados diagramas de secuencia que son usados para modelar la interacción entre los objetos a través del tiempo que componen los sistemas y estos diagramas se modelan para cada caso de uso. En los casos de uso se hace una descripción de la lógica de la aplicación pero en los diagramas de secuencia se abordan detalles de implementación de escenarios como los objetos y las clases dando lugar a mensajes que se intercambian entre los objetos.

De la misma manera que los diagramas de secuencia modelan interacciones entre objetos los diagramas de comunicación además tienen la particularidad de combinar diagramas de casos de usos describiendo partes estáticas y dinámicas del sistema.

2.4.3 Análisis jerárquico de tareas / Hierarchical Task Analysis

Los primeros trabajos sobre Análisis jerárquico de tareas (HTA; (Annett J. & Duncan 1967), 1967; (Diaper D. 2004)) (Análisis jerárquico de tareas) se remontan a finales de los sesenta. La idea básica de HTA es describir el conjunto de actividades que se consideran una estructura lógica en diferentes niveles, ha demostrado ser exitosa, como se puede ver desde su aplicación en un número de proyectos. Sin embargo, este enfoque se describe cómo las actividades están relacionadas entre sí de una manera bastante rudimentaria. La representación se da generalmente como nombres de las tareas en cajas con números que indican el orden de actuación. HTA constituye un aporte sobre la base de funcionamiento en lugar de comportamiento o construcciones psicométricas y el uso de una unidad fundamental llamada operación (operation). Un análisis de las tareas funcional comienza por la definición de metas (goals), antes de considerar acciones por las cuales puede ser cumplida la tarea.

Las tareas complejas se definen en términos de una jerarquía de metas y sub-metas (*subgoals*) anidadas dentro un orden de metas más alto, cada meta, y las sub-metas se representan como una operación. La principal característica de una operación son las condiciones en que se activa la meta y las condiciones que responden a la meta junto con las diferentes acciones que se pueden implementar para alcanzar la meta. Estas acciones podrán ser definidas en términos de sub-metas.

Los principales elementos que conforman el modelo de análisis de tareas HTA son las tareas, metas, la descomposición y la re-descomposición, las operaciones, las acciones, los planes, las reglas de detención de una tarea. Ya se ha hablado de algunas en las definiciones básicas por lo que se hablara posteriormente de las que no han sido abordadas.

La diferencia entre el HTA y otros métodos de análisis de tareas es que en los otros se realiza una lista de actividades y en HTA se identifican las metas de la tarea.

El principal objetivo de HTA es proporcionar un análisis funcional en lugar de una descripción del comportamiento. En rutinas repetitivas las tareas y las acciones varían muy poco, mientras que el ambiente y el propósito permanecen constantes. En tareas más complejas, las mismas metas pueden llevarse a cabo por diferentes vías y medios diferentes, dependiendo de circunstancias propias de cada ocasión. Simplemente listar acciones sin entender para que fueron diseñadas puede ser engañoso. Sistemas complejos, incluidos los relativos a los seres humanos y computadoras, están diseñados con metas definidas, y la comprensión de cómo funciona un sistema alcanza o no sus objetivos designados es el propósito principal del análisis.

2.4.3.1 La descomposición y la re-descripción

Las metas son a menudo complejas, es decir, que se definen por más de un evento o por los valores de más de una variable. Siempre que éstos puedan ser identificados individualmente, el análisis debe especificar estos componentes meta estados en el proceso de *descomposición*. HTA prevé dos tipos de descomposición. La primera comprende la identificación de los meta estados especificados por múltiples criterios, por ejemplo, para llegar a un destino (un evento) habiendo gastado un mínimo esfuerzo (variable 1) y sin lesiones (variable 2).

El segundo tipo de descomposición comprende la identificación de sub-metas en todas las rutas que se pueden tomar para alcanzar el estado meta. Las metas pueden revelar una jerarquía de metas y sub-metas. Este proceso de la descomposición, también conocido como re-descripción, tiene la ventaja que comprende una manera económica la forma de localizar fuentes de error general en el sistema.

2.4.3.2 Reglas de Detención (Stop Rules)

La descomposición en jerarquías de las metas y la re-descripción de las operaciones y sub-operaciones podría continuar indefinidamente sin el uso de una regla de detención que especifica el nivel de detalle.

La regla de detención final es tal que para cuando se detenga se tiene toda la información que se necesita para cumplir con los propósitos del análisis. Sin embargo, desde que el propósito general de HTA es identificar las fuentes de falla en el rendimiento real o potencial una regla de detención general es común: detenerse cuando el producto de la probabilidad de falla (p) y

el costo del fracaso (c) se considera aceptable. Esto se conoce como el criterio de $p \times c$ y su principal beneficio es que se mantiene el trabajo de análisis hasta la cantidad mínima y se centra la atención de los analistas sobre los aspectos de la tarea que son críticos para el éxito general del sistema. La razón para detenerse es que la fuente del error ha sido identificado y el analista puede proponer una solución plausible en términos de diseño del sistema, procedimientos de operación, o formación de los operarios (es decir, mediante el rediseño de la tarea cognitiva).

2.4.3.3 Ejemplo de HTA

La Figura 2-10 describe las metas y sub-metas de un cajero automático (Automated Teller Machine o ATM) usando el paradigma HTA, en el cual se describe el proceso llevado a cabo en la interacción con un cajero automático cuando un usuario requiere retirar dinero desde cualquier auto servicio como un banco.

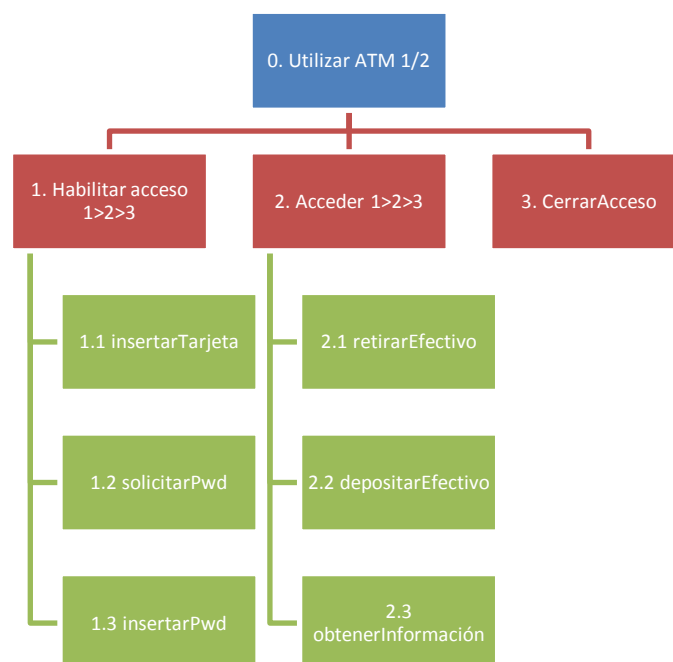


Figura 2-10. Jerarquía de Metas para un ATM

Un eficaz control en la validez de una tabla de descomposición propuesta es que todas las sub-operaciones deben ser mutuamente exclusivas y exhaustivas (es decir, debe definir por completo la operación de orden superior). El uso de la notación estándar en general, ayuda en la construcción de tablas y diagramas así como la comunicación de la interpretación de los resultados, y puede mejorar también entre los analistas y las partes interesadas.

El sistema de notación recomendada ofrece para cada operación identificar un número único que se puede utilizar tanto en diagramas y tablas. La notación también debe especificar planes e indicar detenciones. Detenerse representa el nivel más detallado del análisis, por lo general el nivel que es más relevante para los resultados de la búsqueda y las recomendaciones para continuar la acción.

2.4.4 GOMS (Goals, Operators, Methods, Selection rules)

GOMS (Metas, Operadores, Métodos, reglas de selección) fue el primer aporte en el diseño de interfaces de usuario. GOMS es un método que tiene una larga historia y una considerable influencia. Se basa en un modelo cognitivo que es escrito por un conjunto de memorias y procesadores y un conjunto de principios que subyacen a su comportamiento. Más precisamente, se descompone en tres subsistemas que interactúan entre otros (perceptivo, motor y subsistemas cognitivos).

Un método puede llamar a un sub-método, por lo que los métodos tienen una estructura jerárquica. Si hay más de un método para llevar a cabo una meta, las reglas de selección eligen el método adecuado dependiendo del contexto. Describir las metas, operadores, métodos y reglas de selección de un conjunto de tareas de una manera formal constituye hacer un análisis GOMS, o construir un modelo GOMS.

Una descripción más detallada para describir GOMS es que proporciona una descripción jerárquica para alcanzar las metas en términos de operadores. Los operadores son en esencia actos cognitivos. Las acciones en un nivel pueden ser metas en un nivel inferior. Los métodos son secuencias de sub-metas y los operadores utilizan la estructura de la descripción de cómo llegar a una meta determinada. Las reglas de selección indican que utilizar un método en lugar de uno al otro.

Un ejemplo es cuando se mueve el cursor en una ubicación específica de un documento. Si la posición deseada se encuentra cerca de la actual, entonces es suficiente mover el cursor por las pulsaciones de teclas de flecha, de lo contrario, sería mejor seleccionar la nueva posición con la ayuda del ratón.

Una de las principales características de GOMS es que este modelo de análisis de tareas no es un modelo gráfico, es más bien como un lenguaje de programación que se debe de aprender en contraste con HTA que es un modelo que puede ser representado gráficamente en una tabla o en un árbol de tareas en el que las metas se ven ordenandos por orden de prioridad. Una de las características similares al modelo de tareas HTA es que ambas pueden ser representadas como estructuras jerárquicas de las metas que se desea llevar a cabo en el análisis.

2.4.4.1 Guía del modelo GOMS

A continuación se presenta una guía para poder llevar a cabo un modelo GOMS.

Paso1: Definir los datos de la tarea

La representación de datos básicos en GOMS se compone de objetos con propiedades y valores. Cada objeto tiene un nombre simbólico y una lista de propiedades, cada una de ellas tiene un valor asociado. Los nombres del objeto, la propiedad y el valor son los símbolos. Un ejemplo puede ser un “Desarrollador” que es un oficio y tiene ingresos altos pueden ser representados como se muestra en la Tabla 2-1.

LTM_Item: Desarrollador. Tipo es Informática. Ingreso es alto.
--

Tabla 2-1. Definir datos en GOMS

En este ejemplo, "Desarrollador" es un objeto que tiene una propiedad cuyo valor es "Informática" y otra propiedad cuyo valor es "alto".

Paso 2: Definir las metas

Las metas son, como ya habíamos definido previamente, acciones que los usuarios deben realizar. En este punto de definición de metas podría ser por ejemplo, simulando el ejemplo anterior de un cajero automático, *ACCESAR ATM* como se puede observar es una acción que el usuario debe completar para poder tener acceso a su dinero.

Paso 3: Definir operadores

En las definiciones previas se vio que un operador son acciones que los usuarios ejecutan, en GOMS los operadores pueden ser externos y mentales para ver una lista completa de ellos se pueden encontrar en (Kieras 1994). Así, por ejemplo, retomando el ATM un ejemplo de operador en GOMS se muestra en la Tabla 2-2.

Accomplish_goal: ACCESAR ATM Goto etiqueta_de_paso

Tabla 2-2. Definir operadores en GOMS

El operador en este caso es "Accomplish_goal" el cual define la ejecución de una meta o el cumplimiento de ella.

Paso 4: Definir métodos

Un método es una operación que se debe de realizar al ser llamados por una rutina en alguna parte de un programa. Un método visto de forma más general es una secuencia de pasos que se deben de llevar a cabo para realizar una tarea. Así, por ejemplo en la Tabla 2-3 podemos ver un ejemplo de algunos métodos.

MetodoParaMeta: meta Paso 1. operadores. Paso 2. <operadores> ... Paso i. Accomplish_goal: sub-meta. ... Paso m. Retornar_con_meta_cumplida. MetodoParaMeta: sub-meta Paso 1. operadores. Paso 2. <operadores> ... Paso j. Accomplish_goal: sub-sub-meta. ... Paso n. Retornar_con_meta_cumplida. ...

Tabla 2-3. Método en modelo GOMS

Podemos observar en la Tabla 2-3 que podemos crear tanto métodos como queramos esto es dentro de un método podemos llamar a los métodos que consideremos necesarios para lograr cumplir una meta.

Paso 5: Definir reglas de selección

Las reglas de selección tienen como propósito controlar que los métodos apropiados puedan llevar a cabo una meta. Un ejemplo de reglas de selección lo mostramos a continuación en la Tabla 2-4.

Regla_de_Selection_para_meta: MetaGeneral If (verdadero) Then Accomplish_goal: meta_especifica. If (verdadero) Then Accomplish_goal: meta_especifica. ... Retornar_con_meta_cumplida.

Tabla 2-4. Reglas de selección en modelo GOMS

En la Tabla 2-5 se muestra un ejemplo de una especificación GOMS.

2.4.4.2 Ejemplo GOMS

En la Tabla 2-5 se muestra un ejemplo de un modelo GOMS en el que el dominio de aplicación es un cajero automático ATM (Automated Teller Machine en inglés).

GOAL: ACCESAR ATM GOAL: HABILITAR ACESO INSERTAR TARJETA CREDITO INSERTAR CLAVE GOAL: TOMAR EFECTIVO SELECCIONAR RETIRO SELECCIONAR CANTIDAD DINERO PULSAR OKAY TOMAR DINERO VERIFICAR CANTIDAD DINERO:
--

Tabla 2-5. Ejemplo de un modelo GOMS

En él se describe cómo un usuario realiza la meta a cumplir de acceder al cajero automático *ACCESAR ATM* en una secuencia de operadores pequeños tales como *SELECCIONAR RETIRO* o *SELECCIONAR CANTIDAD DINERO*.

Además se debe de notar que en este ejemplo ninguna regla de selección se utiliza.

2.4.5 La notación ConcurTaskTrees

El Árbol de Tareas Concurrentes o CTT (ConcurTaskTrees en inglés) según (Paternò 2000) es un modelo de tareas desarrollado por Fabio Paterno, un modelo se diferencia del análisis de tareas en que el análisis usualmente se realiza en la fase de verificación de requerimientos y el modelo es la implementación de ese proceso de análisis. CTT consiste de una notación cuyo fin es apoyar los enfoques de ingeniería para el modelado de tareas. El modelo CTT proporciona un rico conjunto de operadores para describir las relaciones temporales entre las tareas lo cual permite concurrencia, secuencialidad y la opcionalidad. Además, para cada tarea la información se puede dar como su tipo, categoría (lo que indica cómo el trabajo está asignado), el objeto que requiere ser manipulado.

El objetivo principal de CTT es especificar interfaces gráficas de usuario mediante el uso de la notación LOTOS. LOTOS es una notación concurrente formal que era una buena opción para especificar las interfaces de usuario, ya que permite a los diseñadores describir los

comportamientos basados en eventos y modificaciones del estado. Aunque LOTOS es una notación ampliamente usada sufre de algunas dolencias o características que la hacen un poco incomoda al trabajar con este tipo de notación ya que hay una necesidad de expresar un conjunto más amplio de comportamientos dinámicos en la interacción persona-máquina de una manera compacta en el análisis y representación de los modelos de trabajo, pero aun así tiene una sintaxis textual que puede generar expresiones complejas, aun cuando el comportamiento de describir es bastante simple.

A continuación veremos algunas de las principales características que se encuentran dentro de la notación CTT.

2.4.5.1 Principales características de CTT

Dentro de las principales características que podemos abordar de la nueva notación CTT encontramos las siguientes:

- **Centrarse en las actividades:** se permite a los diseñadores centrarse en las actividades que los usuarios realizan, que son los aspectos más relevantes en el diseño interactivo aplicaciones que abarcan tanto al usuario como aspectos relacionados con el sistema.
- **Estructura jerárquica:** una estructura jerárquica, característica similar al modelo HTA, parece bastante intuitiva, de hecho, a menudo cuando la gente tiene que resolver un problema que tiende a descomponerse en pequeños problemas que todavía mantiene las relaciones entre las distintas partes de la solución. La estructura jerárquica de esta especificación tiene dos ventajas: que ofrece una amplia gama de granularidad, permitiendo que las estructuras de trabajo grandes y pequeños puedan ser reutilizados, y permite a las estructuras reutilizables de las tareas ser definida en bajos y altos niveles semánticos.
- **Sintaxis Gráfica:** una sintaxis gráfica a menudo, no siempre, es más fácil interpretar, en este caso se refleja la estructura lógica por lo que tiene una forma de árbol.
- **Notación Concurrente:** un rico conjunto de posibles relaciones temporales entre las tareas pueden ser definidas. Este tipo de aspecto esta generalmente implícito, expresado de manera informal en la producción de análisis de tareas. La razón de esta innovación es que después de un análisis de tareas informal lo que sé que quiere es que el diseñador exprese con claridad la lógica de las relaciones temporales. Esto se debe a que tal orden se debería tomar en cuenta en la implementación de la interfaz de usuario para permitir al usuario realizar en cualquier momento las tareas que deberían estar activas desde el punto de vista semántico.
- **Asignación de tareas:** cómo el desempeño de la tarea se le asigna está indicado por la categoría relacionada y se representan explícitamente mediante el uso de iconos. Hay cuatro posibilidades: la tarea de usuario (sólo en la actividad cognitiva interna, tales como la selección de una estrategia para resolver un problema), tarea de la aplicación (sólo el rendimiento del sistema, tales como la generación de los resultados de una consulta); tarea de interacción (las acciones del usuario con la posibilidad de la retroalimentación inmediata del sistema, como la edición de un diagrama), las tareas de resumen (las tareas que tienen sub-tareas que pertenecen a diferentes categorías).
- **Objetos:** una vez que las tareas se identifican, es importante señalar los objetos que tienen que ser manipulados para apoyar su desempeño. Dos grandes tipos de objetos

se puede considerar: los *objetos de interfaz de usuario* y los *objetos de dominio de aplicación*. Varios objetos de la interfaz de usuario se pueden asociar a un objeto de dominio.

2.4.5.2 Operadores Temporales

El modelo CTT se compone de una serie de operadores temporales para poder llevarlo a cabo los cuales se describen a continuación en la Tabla 2-6.

En el operador de habilitación (*Enabling*) una tarea habilita a otra cuando esta termina, un ejemplo del uso de este operador puede ser una base de datos cuando un usuario se debe registrar primero y luego interactuar con los datos. Luego con el operador de desactivación (*Disabling* o Deactivation) la primera tarea (T1) se desactiva una vez que la primera acción de la segunda tarea (T2) ha sido realizada.

Mientras que con el operador temporal de suspensión (*Interruption*) da la posibilidad de interrumpir la primer tarea (T1) cuando la segunda tarea (T2) ha finalizado completamente. El operador de selección (*Choice*) permite escoger entre un conjunto de tareas una vez que la selección ha sido llevada a cabo la tarea escogida puede ser realizada y otra tarea no se habilitara a menos que esta tarea escogida ha sido terminada. Para el operador de recursividad (*Iteration*) significa que el sub-árbol originado por la tarea considerada realiza otra ocurrencia de la tarea es decir esto es un proceso repetitivo el que se realiza.

Luego tenemos al operador de concurrencia (*Concurrency*) que no es más que decir que las acciones pertenecientes a otras dos tareas pueden ser realizadas en cualquier orden sin ninguna limitación específica. Y por último tenemos al operador de tareas opcionales (*Optionality*) que da la posibilidad de indicar que la realización de una tarea específica es completamente opcional.

Descripción	Operador
Enabling	T1 >> T2 o T1 []>> T2
Disabling	T1 [> T2
Interruption	T1 > T2
Choice	T1 [] T2
Iteration	T1* o T1{n}
Concurrency	T1 T2 T1 T2
Optionality	[T]

Tabla 2-6. Operadores temporales en CTT

Una vez que se han abordado las principales características de CTT y se han descrito los operadores temporales a continuación se muestra un ejemplo al igual que en el apartado anterior con el modelo GOMS se retoma la simulación de un cajero automático (ATM) tal y como se muestra en la Figura 2-11.

2.4.5.3 Ejemplo de CTT

La Figura 2-11 presenta un modelo ConcurTaskTrees relativo al cajero automático. El primero es la tarea *HabilitarAcceso* consta de tres sub-tareas secuenciales (>> es el operador de habilitación).

La tarea *RequerirPassword* se realiza por el sistema como se indica con el icono del ordenador. La tarea *Acceder* es una tarea iterativa (indicado por el símbolo *, asterisco), compuesta por la elección entre tres sub-tareas (*RetirarEfectivo*, *DepositarEfectivo*, *ObtenerInformacion*). La tarea *RetirarEfectivo* se compone de una secuencia de tareas ([>> es el operador de cruce que permite el paso de la información). También tiene una sub-tarea de usuario (indicado por un icono similar): la tarea *DecidirCantidad*, que describe la actividad de los usuarios internos cognitivos asociados con decidir la cantidad a retirar.

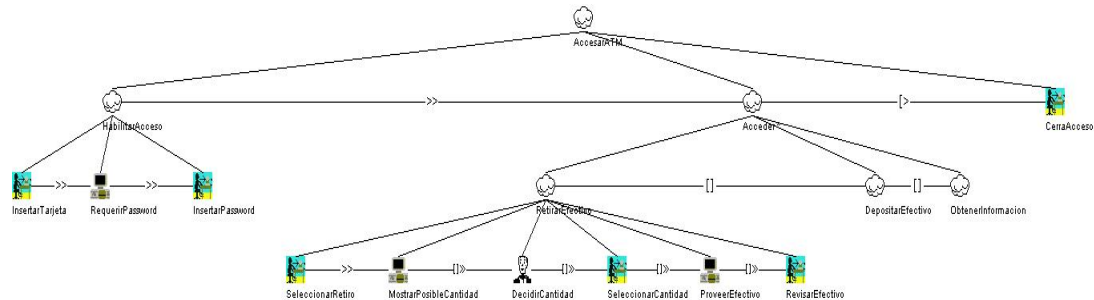


Figura 2-11. Modelo CTT de un ATM

Una vez que hemos abordados algunas de las técnicas de análisis y modelados de tareas más usadas en la sección 2.6 veremos algunos aportes que utilizan algunas de estas técnicas en sus propuestas que abarcan paradigmas de interacción persona-máquina. Aunque estas propuestas hacen uso de técnicas de análisis y modelado de tareas en los paradigmas de interacción no justifican objetivamente por qué deciden utilizarla es decir el proceso de selección de la técnica de tareas es un proceso empírico en los aportes que se abordaran y el aporte de este trabajo consiste en que ese proceso ya no sea empírico sino que se justifique objetivamente por qué se opta por usar una técnica de tareas específica.

2.4.6 UML y sus facilidades para la especificación de tareas

En (Markopoulos 2002) podemos encontrar un aporte en el que hace una comparación entre las técnicas basadas en UML y CTT. Propone extender UML para soportar tareas basadas en el diseño, poblaciones de usuarios en términos de roles y sus relaciones; las tareas son asociadas con roles y se definen como ‘cosas que los usuarios hacen’, las sub-tareas son las mismas tareas que pueden ser asociada con relaciones lógicas y temporales las cuales son definidas como asociaciones entre tareas. En este aporte se asocian tareas que los usuarios realizan con los diagramas de actividad.

2.5 Discusión de las técnicas y notaciones de análisis y modelado de tareas

La discusión realizada en este punto analiza las notaciones descritas con anterioridad e identifica elementos en común y sus diferencias, con el fin de obtener una visión general y comparada de las mismas. Además, se identificarán también en esta sección las diferencias y similitudes que presentan los nuevos paradigmas. Y, por último se lleva a cabo un análisis de los trabajos relacionados con los nuevos paradigmas de interacción y el uso, que en los mismos, se ha hecho de las notaciones descritas. Seguidamente se discutirán, atendiendo a diferentes criterios, las ventajas y debilidades que presentan las notaciones previamente presentadas.

2.5.1 Nuevos paradigmas de interacción vs paradigmas tradicionales de interacción

Una de las características principales entre los paradigmas de interacción existentes (llámese ordenador personal) y los nuevos paradigmas de interacción tiene que ver con el ámbito en el que se realiza esta interacción. Así, por ejemplo, con los nuevos paradigmas lo que se desea es que la interacción entre el hombre y la máquina se lleve a cabo de la forma más natural posible, esto también implica que esta interacción natural se realice para todo tipo de usuarios. Ya que la tendencia que se desea es que la interacción se realice de la forma más natural posible, surge la necesidad de soportar el trabajo colaborativo con las técnicas de análisis y modelado de tareas existentes. Es decir, una de las primeras preguntas que nos hacemos es si las técnicas y notaciones tradicionales siguen siendo válidas para especificar las nuevas exigencias y retos que presenta la interacción persona-ordenador en el ámbito colaborativo.

2.5.2 Diferencia sintáctica (textual vs gráfica)

La diferencia sintáctica es una de las diferencias más evidentes que podemos encontrar entre las notaciones descritas en los apartados de la sección previa. HTA, por ejemplo, es un modelo que requiere ir especificando en cada nivel de definición de metas un número consecutivo seguido por un punto y el nombre de la meta, luego se indica qué meta o metas deben sucederse una vez cumplida esa meta. Así, por ejemplo, para definir la meta cero (*metaCero*) su definición sería (0. *metaCero* 1/2) que indica que después de la meta cero se puede realizar la meta uno (*metaUno*) o la meta dos (*metaDos*). En HTA cada meta debe contener un plan, una acción, entradas y recomendaciones. Los planes son una descripción textual de la actividad realizada. En HTA la representación gráfica es muy rudimentaria, y se lleva a cabo a través de cajas de texto o tablas.

En el modelo GOMS, sin embargo, esto difiere también ya que en este modelo lo que se trata de hacer es tener en cuenta un conjunto de operadores y palabras reservadas tal y como se haría en un lenguaje de programación, por ejemplo Java. Así, para definir una meta y el conjunto de sub-metas con sus operadores y demás se puede revisar la Tabla 2-5 del ejemplo de la sección 2.4.4.2, que describe el conjunto de tareas que se llevan a cabo en un cajero automático a través de una secuencia de operadores y palabras reservadas tal cual se estuviese programando el código fuente de una aplicación.

Con las técnicas de tareas basadas en UML hemos visto que mediante estas se pueden representar los requisitos funcionales, pero no los que no tienen que ver con aquellos aspectos que no tienen que ver con los procesos del sistema. Sin embargo estas técnicas nos permiten representar de forma gráfica las actividades que los usuarios llevan a cabo en la interacción con las aplicaciones y de esta manera brindar una mejor perspectiva del modelo creado.

Por último el árbol de tareas concurrentes, CTT, permite modelar los problemas a través de una herramienta que viene soportada con el paradigma. Con CTT se pueden ver representaciones de las tareas haciendo que esto sea de fácil comprensión para algún diseñador que no esté al tanto de todo el proceso que se llevó a cabo para poder plasmar de la manera más real posible un problema dado. Relacionado también con las facilidades gráficas que ofrece CTT están también los mecanismos para la especificación de tareas que ofrece UML. Un ejemplo significativo de estas facilidades lo encontraríamos en los diagramas de

actividad. En dichos diagramas se puede especificar tanto quién realiza cada tarea, como las tareas que son llevadas a cabo. En el ámbito del desarrollo de interfaces de usuario cabe destacar que CTT se ha venido utilizado con asiduidad en la última década y, sin embargo, hay menos propuestas que hayan tratado de adecuar el uso de las facilidades de UML al desarrollo específico de interfaces de usuario.

2.5.3 Diferentes niveles de formalidad

La diferencia entre los niveles de formalidad es otro aspecto en el que difieren las propuestas que encontramos en la literatura relacionadas con el análisis y el modelado de tareas, que hemos recogido en este documento. Las distintas notaciones difieren en el nivel de formalidad porque los modelos de tareas existentes muestran una gran diversidad en términos de formalidad y profundidad.

Así, por ejemplo, HTA se centra en informar a los diseñadores de los posibles problemas en la utilización del sistema (p.e usabilidad). GOMS se centra en evaluar el rendimiento de las personas, mientras que CTT se centra en soportar el diseño proporcionando un modelo de tareas muy detallado descrito en una jerarquía de tareas. Además, en HTA, por ejemplo, se usa ampliamente el concepto de plan, que es una descripción informal de relaciones temporales, contrastando con CTT, donde las relaciones temporales están explícitamente indicadas por una serie de operadores que se introducen en esta notación.

Por último mencionar que las facilidades ofrecidas por UML para la especificación de tareas son generalistas, no formales y no están directamente enfocadas a la consideración de la interfaz de usuario y su desarrollo. Algunas extensiones, utilizando perfiles, se han propuesto en este sentido donde cabe destacar la propuesta UMLi de (da Silva 2000).

2.5.4 Diferente conjunto de operadores

En HTA como ya se mencionó, una representación informal de operadores se daría a través de un plan. Mientras que en GOMS la definición de operadores vendría dada por las acciones tanto cognitivas y físicas que los usuarios deben de realizar para cumplir una meta específica.

En el árbol de tareas concurrentes, CTT, las tareas están relacionadas a través de un conjunto de operadores temporales diferentes a las dos notaciones anteriores. En CTT se busca con los operadores temporales plasmar las relaciones secuenciales y concurrentes entre el conjunto de tareas especificado. Ese es precisamente uno de los puntos fuertes de esta notación en el que el conjunto de tareas puede ser descrito de forma más precisa y elegante, y no de forma rudimentaria como se hace, por ejemplo, al utilizar HTA.

UML, y sus diagramas de actividad, también ofrece facilidades para especificar relaciones temporales entre tareas, de hecho en (Markopoulos 2002) se identificó la posibilidad de especificar todas las relaciones temporales que ofrece CTT.

2.5.5 Similitudes entre las notaciones de análisis y modelado de tareas

Las similitudes entre las notaciones de modelado de tareas las podemos encontrar en algunos aspectos tales como los que presentamos a continuación.

- Los resultados de utilizar las distintas notaciones son especificaciones que tratan de capturar lo que los usuarios hacen o harían para realizar determinada actividad, es decir, tratan de documentar las actividades que conlleva la resolución de problemas.
- El concepto de ‘paso’: todas notaciones consideradas incluyen una serie de pasos que se deben de realizar para llevar a cabo el proceso de análisis y modelado.
- Herramientas: la totalidad de las notaciones presentadas disponen de herramientas o aplicaciones que dan soporte a las mismas.
- Tareas: en todos los modelos va implícito el concepto de tarea; así, por ejemplo CTT usa esta denominación, mientras que HTA y GOMS se refieren a las tareas como metas.
- Recursividad: las tareas pueden ser descompuestas recursivamente en sub-tareas hasta el nivel de profundidad deseado.

En la Tabla 2-7 podemos ver un resumen de la comparación que hemos realizado hasta este punto.

Similitudes	Diferencias
Modelos abstractos	Sintácticas (textual vs grafica)
Guía de Pasos	niveles de formalidad
Herramientas de apoyo	conjunto de operadores
Descomposición de tareas	

Tabla 2-7 Similitudes y diferencias entre técnicas de análisis y modelado de tareas

2.6 Trabajos relacionados

Además del uso de una notación específica para describir el comportamiento en la interacción en la literatura también se han encontrado aportes en los que se hace uso de técnicas de análisis y modelado de tareas y dar soporte al desarrollo de interfaces de usuario. Una de esas contribuciones la encontramos en (Lozano 2002) en el cual encontramos un aporte metodológico para el desarrollo de interfaces de usuario denominado IDEAS que se basa en la utilización de modelos. Los modelos declarativos que componen el modelo de interfaz de usuario en la propuesta IDEAS son los de casos de uso, de tareas, de usuario, de dominio y de presentación. Además, en IDEAS la relación entre objetos de interfaz y de dominio está basada en los denominados contratos. Un contrato se refiere a que las clases de interfaz no acceden directamente a la información del sistema almacenada en las clases que modelan el dominio, sino que este acceso se hace a través de una clase denominada contrato asociada a la clase de la interfaz. La propuesta metodológica introduce al análisis de tareas como herramienta fundamental en la construcción del modelo de interfaz de usuario, es decir, IDEAS propone un proceso de desarrollo conducido por casos de uso añadiendo técnicas de análisis de tareas basado en UML, para lograr especificar aquellos aspectos relativos a las interfaces de usuario que los casos de uso no permiten abordar, como pueden ser los requisitos no funcionales del sistema.

Podemos encontrar en la literatura existente ejemplos adicionales de desarrollo de interfaces de usuarios basados en modelos que aplican técnicas de análisis y modelado de tareas como veremos a continuación. La notación CTT (Paternò 2000) es utilizada también en (F. Montero 2005), que propone, a partir de esa notación, una propuesta que integra calidad y experiencia

en el proceso de desarrollo de interfaces basada también en modelos y denominada *IDEALXML*. La justificación del autor para el uso de esta notación se basa en que permite potenciar la validación de tareas y considerar en dicho proceso la calidad de la interfaz y la experiencia en su especificación y elaboración. La calidad, tanto funcional como no funcional, puede especificarse utilizando la notación CTT y, a partir de estas especificaciones, puede lograrse la especificación de interfaces de usuario.

Otra aportación que utiliza CTT la encontramos en (Lopez-Jaquero 2005) donde se propone una ampliación de los métodos tradicionales basados en modelos para el diseño de interfaces de usuarios con mecanismos de adaptación. Los mecanismos de adaptación propuestos sirven tanto para soportar el propio proceso de desarrollo de interfaces de usuario como para dotar a la interfaz de usuario generada de capacidad para considerar al usuario, a la plataforma o al contexto en que dicha interfaz es utilizada. En esta propuesta se utiliza CTT, pero no hay una justificación clara y definitiva de esta elección.

Dentro del ámbito del paradigma de entornos colaborativos conocidos como CSCW (Computer Supported Collaborative Work) encontramos la propuesta recogida en (Ruiz 2007). En esta propuesta se relaciona el paradigma de entornos colaborativos con el análisis y modelado de tareas, que el autor denomina TOUCHE (Task-Oriented and UserCentered Process Model for Developing Interfaces for HumanComputer- Human Enviroments). El autor define un modelo conceptual basado en cuatro modelos para la especificación de un sistema colaborativo. En primer lugar, la estructura organizativa de los usuarios de un entorno colaborativo se representa mediante el Modelo de Organización; mediante este modelo se representa la estructura de los actores del sistema así como las relaciones de colaboración que tienen lugar entre ellos. El segundo modelo propuesto es el Modelo de Tareas, que especifica las acciones individuales o colectivas que deben realizar los usuarios del sistema para conseguir objetivos determinados y está basado en CTT. El tercer modelo que se propone es el Modelo de Objetivos, que representa las metas de los grupos y los fines para los que se llevan a cabo las tareas a diferente nivel de granularidad. El último modelo propuesto es el Modelo de Sesión, que proporciona una vista del sistema en un momento determinado y es la representación de un escenario concreto. En la Figura 2-12 se muestra la relación entre los modelos propuestos en (Ruiz 2007).

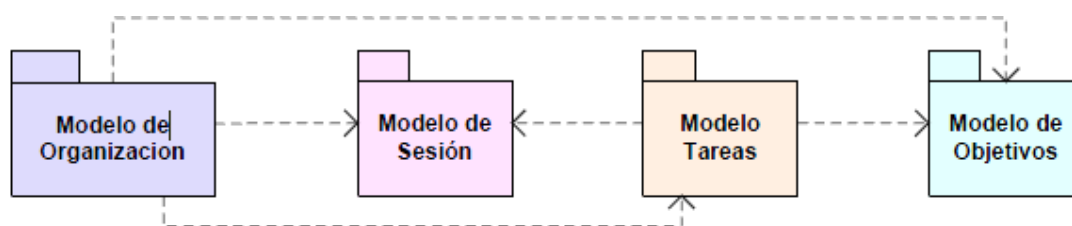


Figura 2-12 Relación entre modelos en un entorno CSCW

(fuente: Ruiz, 2007)

En (Molina 2007) se presenta una propuesta que relaciona al paradigma de interacción de realidad virtual (VR) y al análisis y modelado de tareas. Esta propuesta se denominada TRES-D y se basa en tres meta-modelos (uno para el objeto, uno para los elementos de interacción, y el último para el espacio) útiles para proporcionar un lenguaje para los diseñadores, y un

modelo de proceso que estructura el propio desarrollo. La metodología TRES-D aporta dos enfoques diferentes para el desarrollo de interfaces de usuario, uno orientado a las tareas y el otro orientado al contenido. La metodología Tres-D ayuda al programador a la hora de transformar interfaces existentes a entornos tridimensionales y se complementa con la librería VUIToolkit, que transforma los widgets en los planos en 3D mediante la adición de profundidad a ellos. En esta propuesta se utiliza la técnica de tareas HTA porque el autor argumenta que en entornos virtuales las tareas suelen reducirse a una simple lista de operaciones siendo esto lo que se persigue y con HTA se esgrime que se puede capturar mejor la dimensión temporal de las tareas y de las operaciones.

Otro aporte en el ámbito de entornos virtuales colaborativos (CVEs) lo encontramos en (García 2010). En esta propuesta se proponen tres mejoras para dar soporte a la interacción en CVEs. La primera propuesta consiste en adaptar la interacción a los dispositivos que se utilicen, intentando no reutilizar el diseño de dicha interacción entre sistemas que hagan uso de distintos dispositivos, en síntesis plantea la evaluación de la usabilidad de un CVE que se ejecuta sobre distintas plataformas de hardware, abordando el problema de la diversidad existente y su influencia en la interacción. En la segunda propuesta se abordan el problema que introducen las redes de comunicaciones de las que es común servirse en los CVEs tratando de comprobar el grado en el que afecta la latencia de la red a las técnicas de gestión de la consistencia y a la experiencia del usuario con el sistema. Y, finalmente, el tercero trata la colaboración a un nivel más alto, centrando la atención en cómo los usuarios perciben el mundo compartido y a las formas de ayudar a que comprendan mejor lo que sucede en el mismo. Este mismo autor hace uso de la técnica de modelado de tareas HTA ya que considera que las tareas que los usuarios llevarán a cabo son normalmente compuestas por lo que se pueden descomponer en otras tareas más específicas. Propone HTA porque considera que se realiza un estudio sencillo dando la posibilidad de usar otras técnicas de modelado, como CTT, por lo que podemos notar que también su uso no está justificado objetivamente sino que queda sujeto al conocimiento que tiene sobre una notación específica u otra.

De esta manera hemos notado que para los mismos entornos virtuales colaborativos hemos visto dos notaciones seleccionadas para el mismo paradigma de interacción (HTA y CTT) lo cual nos da la suficiente evidencia para afirmar que no existen criterios objetivos para decantarse por alguna técnica en este ámbito sino que, como ya hemos dicho, está sujeta al criterio de quien realiza el modelado.

A continuación en la Tabla 2-8 presentamos un resumen de los aportes que hemos descrito en esta sección y las técnicas que se han abordado en los paradigmas de interacción que hemos descrito.

Aporte	Técnica	Propósito
IdealXml	CTT	Potenciar validación de tareas y considerar en dicho proceso la calidad de la interfaz y la experiencia
IDEAS	Técnicas de tareas basadas en UML	Especificar requisitos no funcionales
TOUCHE	Basadas en UML y CTT	Representar flujos de trabajo, representar metas de grupos y fines a cualquier nivel

Aporte	Técnica	Propósito
		de granularidad
TRES-D	HTA	Capturar mejor la dimensión temporal de tareas y operaciones

Tabla 2-8 Resumen de técnicas en aportaciones

2.7 Conclusiones

El análisis y modelado, y concretamente las notaciones dedicadas al modelado de tareas, como hemos comentado en este capítulo, tiene una rica y gran variedad de conceptos e implicaciones asociadas. En este capítulo hemos identificado un grupo de técnicas y notaciones asociados al análisis y modelado de tareas que se utilizan también para dar soporte al desarrollo de interfaces de usuario.

Concretamente, en este capítulo hemos identificado y caracterizado las notaciones HTA, GOMS, CTT y aquellas basadas en UML. Dichas notaciones han sido utilizadas de una manera no siempre justificada en el desarrollo de interfaces de usuario basadas en modelos. Identificándose similitudes y diferencias entre ellas.

En este último sentido, además de la faceta sintáctica, en este capítulo hemos identificado diferencias en cuanto al nivel de formalización de las distintas notaciones ya que, por ejemplo, HTA usa el concepto de plan como una manera formal de describir las relaciones temporales entre las tareas y otras notaciones, como CTT, tiene una notación bien definida para tal fin, que está basada en LOTOS.

Junto a las diferencias también se han identificado elementos comunes en las notaciones estudiadas, entre ellos cabe destacar que todas las notaciones estudiadas sugieren realizar un conjunto de pasos para poder realizar el modelado y tratan de facilitar herramientas que faciliten la especificación de los modelos realizados.

Este capítulo ha perseguido identificar el análisis y discusión de algunas de las más habituales y recurridas notaciones de análisis y modelado de tareas, pero ese no era el único propósito. Otro factor deseaba ser considerado en esa discusión. Dicho factor son las necesidades y la evolución que previsiblemente tienen y tendrán las interfaces de usuario en estos momentos y cómo las técnicas de análisis y modelado de tareas actuales están en disposición de hacerles frente. En este sentido, resaltamos la necesidad de desarrollar interfaces de usuarios de tal manera que la interacción se realice de la manera más natural posible entre los usuarios y las máquinas, es decir tomando como principales elementos del diseño el usuario y las actividades que éste realiza ya que son estos factores muy importantes en el proceso de desarrollo de sistemas.

En el próximo capítulo se analizarán las distintas notaciones descritas en este capítulo, y se hará mediante una serie de criterios que se identificarán como necesarios en función de la evolución previsible de las interfaces de usuario.

Capítulo 3

Trabajo de Investigación: Evaluando las técnicas de análisis y modelado de tareas considerando distintos criterios de interacción persona-ordenador

En este capítulo se describirán y mostrarán los datos obtenidos de un trabajo de investigación incluido dentro de la propuesta de tesis doctoral. Concretamente, en este trabajo de investigación, primero se identificarán una serie de requisitos y necesidades que se están demostrando como necesarias para el desarrollo de las interfaces de usuarios actuales y previsibles y después se analizará el soporte que las notaciones de modelado y análisis de tareas, consideradas en el capítulo anterior, ofrecen a dichos requisitos.

Por tanto, el principal resultado de este trabajo de investigación es identificar la idoneidad y, en su caso, las limitaciones o bondades que presentan las principales notaciones de análisis y modelado de tareas disponibles desde el punto de vista del desarrollo de interfaces de usuario. Las notaciones de análisis y modelado de tareas que serán consideradas serán las identificadas en el capítulo anterior.

Este trabajo de investigación está organizado atendiendo a las siguientes secciones. En primer lugar identificaremos cuáles son nuestras entradas, es decir, por un lado las notaciones de modelado de tareas elegidas y, por otro, la identificación de la técnica de evaluación para abordar nuestro análisis. La técnica de evaluación seleccionada se basa en la propuesta DESMET (Kitchenham 1996) integrada dentro de un método de Investigación-Acción (Baskerville R. 1999). En la segunda sección se identificarán los principales criterios utilizados para realizar la evaluación de las notaciones consideradas. Describiremos, posteriormente, cuándo, cómo, quién y en qué ha consistido la evaluación llevada a cabo. Finalmente, se recopilarán, analizarán y discutirán los resultados obtenidos y se propondrán trabajos futuros.

3.1 Análisis empírico de técnicas de análisis y modelado de tareas en diferentes tipos de plataforma y modalidad

Este trabajo de investigación se llevará a cabo evaluando las técnicas y notaciones de modelado de tareas descritas en el estado del arte recopilado en el capítulo anterior, es decir, HTA, GOMS, CTT y las basadas en UML. El análisis partirá del establecimiento de una serie de características o criterios que debería tener, al menos, una notación ideal para cubrir los aspectos de análisis y modelado de tareas para las interfaces de usuario actuales y futuras. Conviene recalcar que las notaciones mencionadas con anterioridad se propusieron, cuando menos, hace quince años y que las interfaces de usuario han cambiado mucho en ese mismo período temporal, por ello el trabajo que se propone, a nuestro juicio, está plenamente justificado.

La hipótesis de partida de este trabajo de investigación es que las notaciones no cubren todas las necesidades y requisitos que sugiere la evolución de las interfaces de usuario, aunque, sin embargo, algunas de las notaciones estará mejor posicionada para cubrir dichas necesidades.



Figura 3-1. Método Investigación-Acción y DESMET

Para lograr contrastar dicha hipótesis utilizamos el método Investigación-Acción y la metodología DESMET (véase Figura 3-1) para guiar la evaluación. La utilización conjunta de los métodos mencionados se debe a la naturaleza empírica de la fase de evaluación del método Investigación-Acción (véase descripción resumida de sus fases y acciones en Tabla 3-1), y a la necesidad de reforzarlo con otras metodologías, como DESMET, la cual, según (Kitchenham 1996), ayuda al evaluador de una organización a planificar y a ejecutar la evaluación de metodologías y herramientas de manera imparcial y fiable.

Fase	Acciones
Diagnosticar	Corresponde a la identificación de los principales criterios y necesidades que atañen a la organización y que motivan su deseo de cambiar. Engloba todo lo referente a la interpretación del problema considerado atendiendo a distintas vistas.
Planificar la acción	Esta actividad especifica las acciones organizacionales que deberían tomarse para considerar los criterios establecidos. El descubrimiento de los planes de acción es guiado por el marco teórico, el cual indica el estado futuro deseado por la organización, y los cambios requeridos para alcanzar ese estado. El plan establece el objetivo del cambio y el enfoque para cambiar.
Tomar la acción	Implementa el plan de acción. Los investigadores y participantes colaboran en la intervención activa dentro de la organización cliente, provocando ciertos cambios. Diversas estrategias de intervención pueden ser adoptadas de forma directiva (los investigadores dirigen el cambio), no directiva y táctica.
Evaluar	Una vez completadas las acciones, los investigadores y demás participantes evalúan las salidas. La evaluación considerada en el método Investigación-Acción es iterativa. Por cuestiones de tiempo las modificaciones realizadas en esta fase y en este trabajo de investigación han pasado por la consideración de DESMET.
Especificar el aprendizaje	A partir del resultado de la evaluación, los investigadores especifican el conocimiento adquirido y analizan las conclusiones alcanzadas.

Tabla 3-1 Fases y acciones del método Investigación-Acción

Cuando en la Tabla 3-1 **¡Error! No se encuentra el origen de la referencia.** se hace referencia a investigadores y participantes se quiere hacer referencia a los involucrados en las actividades llevadas a cabo para tratar de contrastar la hipótesis nula de partida comentada anteriormente. Aunque los participantes en el proceso de evaluación serán objeto de descripción detallada más adelante, por ahora indicar que su perfil responde a personal docente e investigador relacionado con la disciplina de Interacción Persona-Ordenador. Eso no quita para que algunos de esos investigadores estén familiarizados con otras disciplinas, como por ejemplo Ingeniería del Software.

3.2 Evaluación de técnicas de análisis y modelado de tareas en paradigmas post-WIMP

En esta sección se identificará cómo se ha puesto en práctica el proceso de evaluación utilizando DESMET. Es decir, se identificarán: entradas, criterios de evaluación, participantes, mecanismo de evaluación y sondeo, y resultados que deberían obtenerse. En secciones posteriores se analizarán dichos resultados.

3.2.1 Evaluación de las técnicas y notaciones de análisis y modelado de tareas

En esta sección se presenta el procedimiento llevado a cabo para realizar el proceso de evaluación de las notaciones de modelado de tareas.

3.2.1.1 Procedimiento de evaluación

El primer paso consistió en escoger la muestra de nuestros experimentos, estos fueron un grupo de nueve expertos; siete del área de interacción persona ordenador y dos de ingeniería de software. Seguidamente se establecieron una serie de criterios que se presentan en el desarrollo de interfaces de usuario actuales y futuras.

Posteriormente se realizó un experimento aplicando la metodología DESMET que permitió establecer el grado de importancia de los criterios que habíamos definido anteriormente por el grupo de expertos. Luego, en el segundo experimento, continuando con la metodología de evaluación, se procedió a evaluar a cada una de las notaciones; las técnicas basadas en UML fueron evaluadas por el grupo de expertos de ingeniería de software y las notaciones de modelado de tareas restantes fueron evaluadas por el grupo de expertos de IPO.

Y por último, se procedió a realizar un análisis de los resultados obtenidos de la evaluación de forma global.

3.2.1.2 Participantes en la evaluación

El primer paso consistió en clasificar la muestra de nuestros experimentos; en este caso se escogieron dos grupos atendiendo a las características que presentan cada uno de ellos. El primer grupo está formado por expertos en el área de ingeniería de software pertenecientes al grupo de investigación ISE Research Group de la Universidad UCLM en el campus de Albacete. El segundo grupo seleccionado fue a expertos en interacción persona ordenador de la misma Universidad pertenecientes al grupo LoUISE con una amplia experiencia en el uso de notaciones de modelado de tareas.

3.2.1.3 Paso 1: determinación de los criterios de evaluación

Los criterios considerados para identificar limitaciones o capacidades de las notaciones utilizadas para modelar y analizar tareas se han seleccionado atendiendo a la evolución presente y previsible que tendrán las interfaces de usuario a través de especificaciones o características que debería de tener una técnica/notación de análisis y modelado de tareas. La metodología que se ha seleccionado para evaluar estas técnicas y notaciones es DESMET que nos permite evaluar técnicas de ingeniería de software. Esta técnica de evaluación nos permite realizar estudios de carácter cualitativo y cuantitativo a través de un caso de estudio. El tipo de estudio que llevaremos a cabo será tanto cualitativo como cuantitativo, ya que ambos tienen puntos en común y nos ayudarán a reforzar nuestra hipótesis sobre el problema planteado.

La Tabla 3-2 muestra el listado de las principales características o criterios que debería de tener una técnica/notación de análisis de tareas para que cumpla con las necesidades de independencia de modalidad y plataforma.

Criterio	Descripción
Facilidad para la especificación de productos software altamente interactivos	La notación debe permitir especificar la interacción entre usuario y computador haciendo uso de productos software altamente interactivos (p.e alto grado de variabilidad de las tareas en software para juegos).
Facilidades para extender, mantener y gestionar especificaciones escalables de modelos de tareas	La notación debe ser extensible o manejable (p.e la notación no debe crecer sin control, además debe permitir agregar características especiales como operadores temporales para mantener un grado de formalidad)
Capacidad para poder especificar diferentes estilos de interacción y considerar características del usuario (accesibilidad)	La notación debe de ser independiente al estilo de interacción (p.e línea de comandos, menú, Form Fillin, etc.) y además deberá permitir la especificación de tareas para todos los usuarios
Capacidad para simular, ejecutar y barajar diferentes alternativas en la especificación total o parcial de los modelos de tareas	La notación debe permitir simular diferentes alternativas a través de especificación de diferentes tipos de tareas
Facilidades de integración de las especificaciones realizadas utilizando diferentes notaciones	La notación debe permitir la integración con otras notaciones existentes al modelar tareas o sub-tareas
Posibilidad de identificar y definir especificaciones parciales y reutilizables	La notación debe permitir crear sus propios macros basados en la concepción de sus propios dominios para realizar tareas
Posibilidad de integración con la experiencia	La notación debe permitir la integración con la experiencia a través de patrones de interacción
Almacenamiento en lenguajes estándar	La notación debe permitir el almacenamiento en lenguajes estándar (p.e lenguajes basados en XML y relacionados con la especificación de interfaces de usuario)

Criterio	Descripción
Soportar especificaciones relacionadas con awareness	La notación debería ser capaz de facilitar la especificación de los conocimientos que el usuario debería disponer, o ser capaz de interpretar, para realizar las actividades y cubrir los objetivos que desea.
Soportar especificaciones relacionadas con el feedback	La notación debería ser capaz de permitir la especificación de diferentes tipos de feedback o retroalimentación, que el sistema debería proporcionar para indicar su estado y el de las tareas realizadas y posibles.
Útil para soportar actividades de evaluación de la interacción realizada por los usuarios	La notación debería ser capaz de permitir dar soporte a actividades de evaluación de las actividades llevadas a cabo por los usuarios. Facilitando las posibilidades de comparar la especificación y las acciones reales llevadas a cabo.
Evaluación de la calidad de los propios modelos de tareas elaborados (complejidad, facilidad de entendimiento, modificabilidad/mantenimiento, etc.)	La notación debería disponer de facilidades para evaluar la calidad de las especificaciones realizadas.
Facilidades de documentación y descripción de las especificaciones realizadas	La notación debe permitir una fácil comprensión del modelo creado a través de la documentación y descripción para ser entendible fácilmente
Ofrecer soporte a la realización de tareas tanto cooperativas como colaborativas. Dando la posibilidad de especificar cómo se soportan las actividades de colaboración, coordinación, comunicación y compartición de la información.	La notación debe soportar tareas que se llevan a cabo por más de un usuario en diferentes entornos describiendo las actividades a través de descripciones de forma detallada y fácilmente manejable

Tabla 3-2 Lista de características de una técnica/notación general de análisis y modelado de tareas

La facilidad que se debe tener en la especificación de productos software altamente interactivos es un criterio vital en el desarrollo de interfaces de usuario y más aún en el desarrollo de software cuyo ámbito es el entretenimiento ya que p. e. es difícil modelar con las técnicas/notaciones existente el alto grado de variabilidad en juegos en donde el usuario pueda interactuar libremente a través de un gran conjunto de tareas y especificar a un nivel de detalle manejable las sub-tareas involucradas en el proceso.

Además de la facilidad en la especificación de la interactividad, debe existir la facilidad para extender, mantener y gestionar especificaciones de tareas haciéndolas escalables. Esta característica facilitaría la incorporación no traumática de nuevas tareas en una especificación dada y el uso de experiencia documentada y disponible.

La capacidad que debe tener la técnica/notación de tareas para poder especificar diferentes estilos de interacción como los abordados en el estado del arte del capítulo anterior (p.e la manipulación directa) y considerar características del usuario resulta también importante ya que es muy común, hoy en día, ver interfaces en las que los usuarios seleccionan objetos y los arrastran provocando cambios en la interfaz subyacente. Un ejemplo de esto nos lo brindan los teléfonos móviles táctiles en el que el usuario interactúa con la interfaz a través de la selección de objetos y estos objetos provocan cambios en el estado de la interfaz.

La técnica/notación de tareas debe de permitirnos simular diferentes alternativas en la especificación total o parcial de los modelos de tareas a través de la especificación de diferentes tipos de tareas realizadas en algunos casos por la máquina y en otros por los usuarios. Un ejemplo de esto lo podemos encontrar en la interacción que se da en los pacientes con el síndrome de encierro³ los cuales interactúan a través de un casco intentando mover un cubo con el 'pensamiento'. De esta manera la interacción se representaría por ejemplo con una tarea estilo hombre-máquina pero con la idea de interfaz invisible la interacción se realizaría de tal manera que el usuario podría en cualquier instante interactuar de una manera diferente para realizar una misma tarea.

La facilidad de integración de las especificaciones realizadas utilizando diferentes notaciones es referida a que la técnica/notación de análisis y modelado de tareas debe permitir integrar otras técnicas/notaciones que ya hayan sido propuestas, de esta manera podríamos modelar ciertas tareas con nuestra propuesta y luego algunas de las sub-tareas de ella podrían ser modeladas con una de las notaciones existentes que puedan representar mejor esa actividad mientras se realiza una posible extensión que abarque las características que aún no satisface la notación propuesta.

Otra de las características que debe de tener la técnica/notación de tareas es la posibilidad de identificar y definir especificaciones parciales y reutilizables ya que podríamos construir macros basados en la concepción que tienen los usuarios de sus propios dominios. Esto lo podemos ver en interfaces de usuarios que asociamos a redes sociales, como Facebook, en la cual los usuarios crean sus propias páginas web personalizadas a partir de su perfil.

La posibilidad de integración con la experiencia es importante desde el punto de vista de la disponibilidad de experiencia de diseño documentada utilizando diferentes formatos. Por ejemplo mediante patrones. Este criterio está especialmente relacionado con la escalabilidad, citado con anterioridad. En la actualidad existen innumerables publicaciones que documentan y difunden experiencias de diseño exitosas. Sin embargo pocas incluyen modelos o especificaciones independientes de la plataforma de estas experiencias. La disponibilidad de una notación de análisis y modelado de tareas como la que perseguimos podría utilizarse en este ámbito.

El soporte de awareness (en CSSW) es importante ya que esta característica permite a los usuarios en entornos colaborativos tener conocimiento o ser conscientes de los eventos o

³ http://www.bbc.co.uk/mundo/noticias/2011/10/111010_poder_mente_mover_objetos_mr.shtml.

tareas que se están llevando a cabo por otros usuarios de forma ‘síncrona’. Un ejemplo de la importancia del awareness lo encontramos en GoogleDoc en el cual los usuarios pueden editar simultáneamente un documento y el conocimiento de que usuario está editando que párrafo o qué página nos lo ofrece el ‘Nombre de usuario’ representado por un color diferente por cada usuario que está accediendo al documento. Además se puede tener una vista de todos los usuarios que están haciendo uso de él, este feedback contribuye al awareness o consciencia que desarrolla un usuario de dicha aplicación.

El feedback (complementario al awareness) debe de estar presente para informar a los usuarios de los cambios que acontecen en el producto software tras la ejecución de una tarea dada. Por ejemplo, un ejemplo de feedback lo encontramos en un ATM en el cual por cada tarea ejecutada por el usuario (ejemplo ingresar clave) el sistema le responde a través de un mensaje en pantalla (proporcionándole feedback sobre si fue correcta o incorrecta su interacción) y así sucesivamente hasta retirar la cantidad de dinero deseada.

La facilidad de documentación y descripción de las especificaciones realizadas permitirá tener una mejor concepción en la descripción del modelo de tareas ya que será fácil de entender por diseñadores que empiezan a trabajar en un modelo ya desarrollado.

El soporte a tareas colaborativas/cooperativas debe estar soportado ya que hoy día se desarrollan aplicaciones en las que están inmersos una gran cantidad de usuarios y esto supone un grado de complejidad en su representación. Ejemplo de esto lo encontramos en un controlador de versiones (SVN) de desarrollo de software en el cual se trabaja de forma colaborativa y se lleva un control del diseño, el código, el momento y el responsable de cada acción entre otras informaciones.

3.2.1.4 Paso 2: evaluación comparativa de notaciones de A&M de tareas

De acuerdo con DESMET, después de haber listado un conjunto de características que se quieren utilizar para evaluar las notaciones de análisis y modelado de tareas, lo siguiente es asignar un grado de importancia a las características identificadas estos grados de importancia se muestran en la Tabla 3-3.

Definidos los grados de importancia, el método cualitativo nos dice que debemos de asignar a cada característica el grado que consideremos mejor tal y como se muestra en la Tabla 3-4.

Representación	Descripción
A	Imprescindible
B	Muy deseable
C	Deseable
D	Recomendable

Tabla 3-3 Lista de grado de importancia usando DESMET

Cabe destacar en este punto que los valores representados por la columna ‘Grado de importancia’ de la Tabla 3-4 se obtuvieron del primer cuestionario realizado a un grupo de expertos, tanto de ingeniería de software como de interacción persona ordenador, en el cual se les pidió ponderar, según su criterio, el nivel de importancia que ellos daban a esos criterios, y de esta manera llegar a un consenso entre todos los involucrados. La Figura 3-2 nos muestra

los resultados que hemos obtenido indicándonos el valor que hemos descrito anteriormente; Imprescindible (4), Muy deseable (3), Deseable (2), Recomendable (1).

Criterio	Grado de importancia	Alias
Facilidad para la especificación de productos software altamente interactivos	B	Int.
Facilidades para extender, mantener y gestionar especificaciones escalables de modelos de tareas	B	Esc
Capacidad para poder especificar diferentes estilos de interacción y considerar características del usuario (accesibilidad)	B	Est
Capacidad para simular, ejecutar y barajar diferentes alternativas en la especificación total o parcial de los modelos de tareas	B	Sim
Facilidades de integración de las especificaciones realizadas utilizando diferentes notaciones	B	Not
Posibilidad de identificar y definir especificaciones parciales y reutilizables	B	Reu
Posibilidad de integración con la experiencia	C	Exp
Almacenamiento en lenguajes estándar	D	Alm
Soportar awareness	A	Awa
Soportar feedback	A	Fee
Soportar actividades de evaluación de la interacción realizada por los usuarios	B	Eva
Evaluación de la calidad de los propios modelos de tareas elaborados (complejidad, facilidad de entendimiento, modificabilidad/mantenimiento, etc.)	C	Cal
Facilidades de documentación y descripción de las especificaciones realizadas	B	Doc
Ofrecer soporte a la realización de tareas tanto cooperativas como colaborativas. Dando la posibilidad de especificar cómo se soportan las actividades de colaboración, coordinación, comunicación y compartición de la información.	B	Coo

Tabla 3-4 Nivel de importancia de las características de técnica/notación general de análisis y modelado de tareas

La Figura 3-3 muestra que el 67% de los expertos pertenecen a la disciplina de interacción persona ordenador lo cual nos demuestra que la mayoría de ellos tienen más afinidad con esa disciplina que con la ingeniería de software u otra. El 11% de nuestro grupo de control es exclusivamente de ingenierías de software y la mayoría que indicaron otros (22%) como disciplina hacen referencia a que manejan ambas disciplinas.

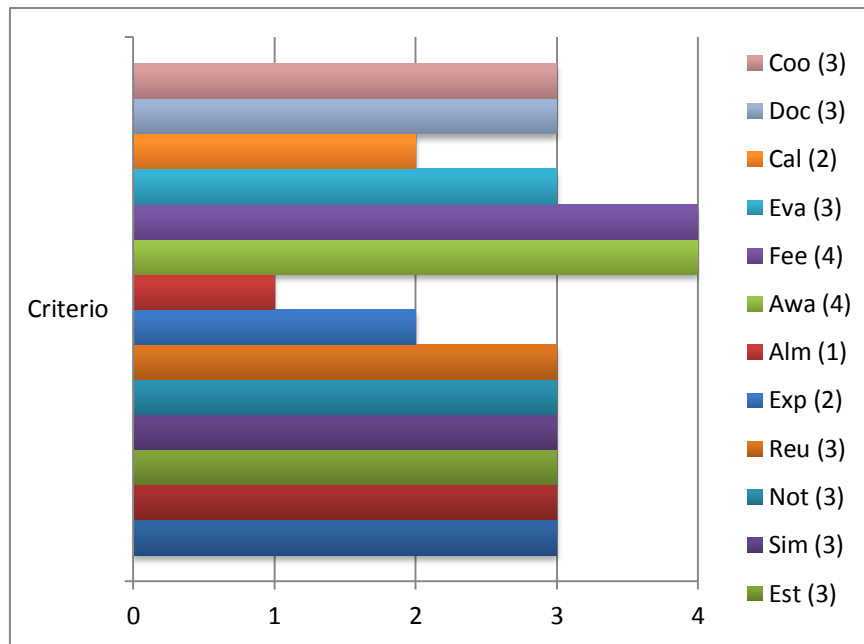


Figura 3-2 Nivel de importancia obtenido según el grupo de expertos

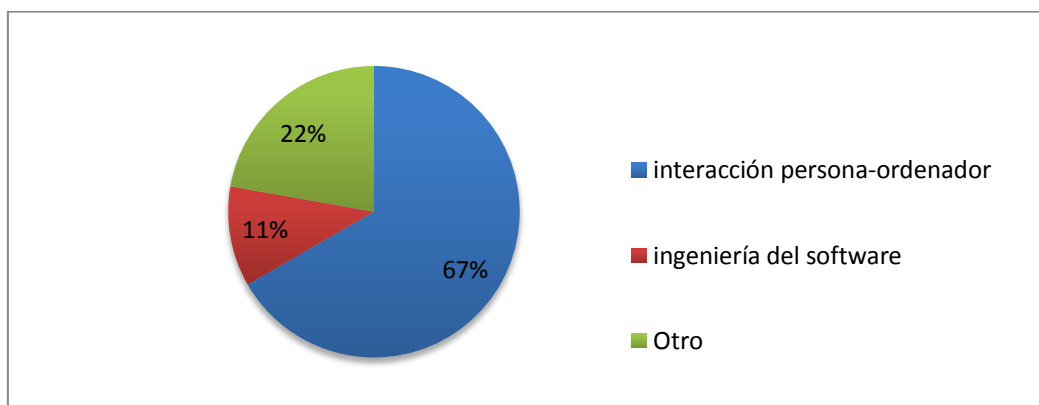


Figura 3-3 Disciplinas con la que guardan más afinidad los expertos

Analizando los datos presentados en la Figura 3-3 podemos afirmar que el consenso alcanzado por los valores del nivel de importancia que se presentan en la Tabla 3-4 están debidamente justificados ya que la mayoría de los expertos (67%) viene de la disciplina IPO quienes son los que más trabajan con las técnicas/notaciones de análisis y modelado de tareas.

La Figura 3-4 muestra las notaciones de análisis y modelado de tareas más conocidas o más utilizadas por el grupo de expertos consultados en este análisis, lo que nos permite darnos cuenta de cuáles de ellas se dispone de más experiencia. De esta manera podemos ver en Figura 3-4 que las técnicas de modelado de tareas basadas en UML y la notación CTT son las más conocidas, con un 100% y un 87,5% respectivamente. Otras notaciones como HTA y GOMS tienen menor uso o aceptación entre los participantes de nuestra consulta.

En tercer lugar esta HTA con un 62,5% y esto es debido a que la mayoría de los expertos no se identifican con esta notación ya que es una de las más antiguas, además que no ofrece las ventajas que se pueden tener con CTT, y en último lugar esta GOMS con un 37,5% la cual es uno de los primeros aportes en IPO y esto lo hace una notación obsoleta y muy poco conocida

y además por ser uno de los primeros aportes en IPO esta no modela de forma ideal los nuevos requerimientos que han ido surgiendo en el modelado de tareas tales como algunas de las características o criterios que hemos presentado.

La Figura 3-5 refleja la experiencia y tiempo que conocen los expertos consultados con las técnicas/notaciones de modelado de tareas, indicándonos que la mayoría tienen más de dos años de experiencia (un 67%) por lo cual el nivel de importancia está debidamente justificado ya que la mayoría de los expertos consultados en el primer cuestionario tienen amplia experiencia en las notaciones que hemos presentado. De la misma grafica podemos deducir que algunos de los expertos tienen menos de dos años (un 11%) lo cual es un porcentaje bajo considerando el anterior. Y por último, la cantidad de expertos que tienen menos de un año de experiencia con las técnicas/notaciones de análisis y modelado de tareas es de un 22%, no existiendo ninguno de los participantes consultados que no posea experiencia con dichas notaciones.

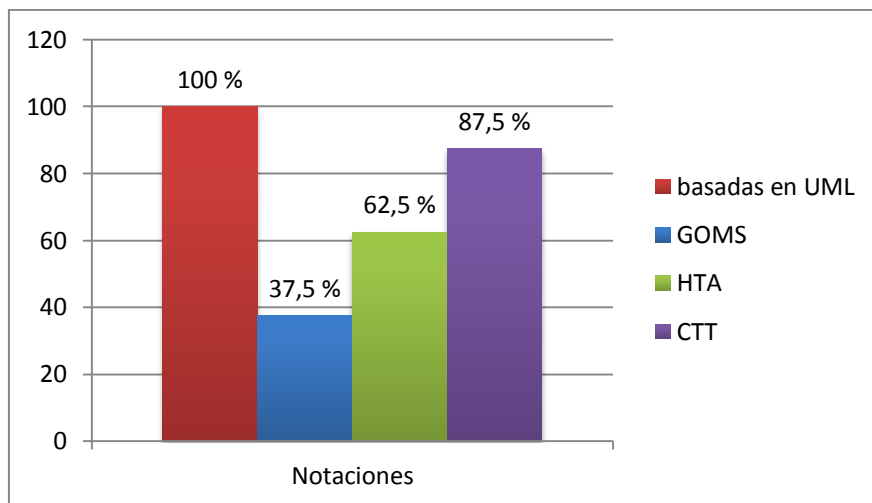


Figura 3-4 Notaciones de análisis y modelado de tareas con más experiencia

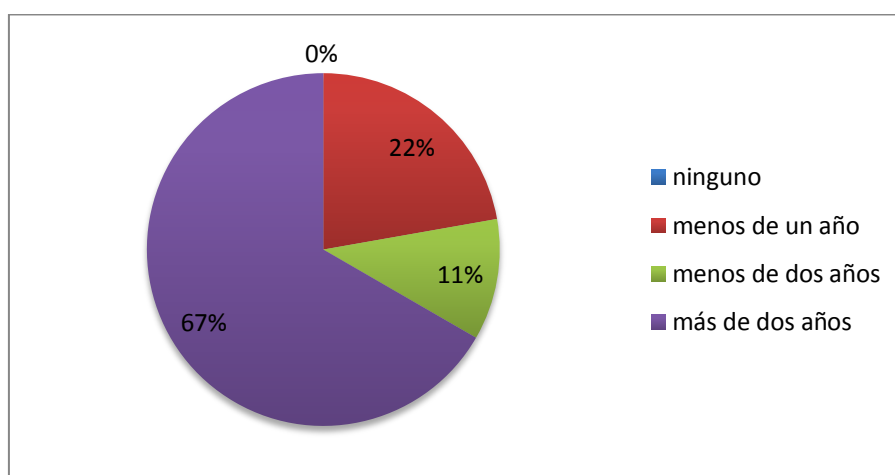


Figura 3-5 Tiempo de experiencia en notaciones de modelado de tareas

Posteriormente se procedió a realizar un segundo experimento en el que se les pedía al grupo de expertos de ambas disciplinas evaluar con qué grado las notaciones existentes soportan los criterios que hemos definido anteriormente.

Una vez que se ha asignado el nivel de importancia a cada característica, según DESMET se procede a proporcionar una escala de juicio para evaluar las técnicas/notaciones de análisis y modelado de tareas que soportan las características descritas anteriormente. La Tabla 3-5 muestra las escalas que se usaran para evaluar las características. Las escalas se utilizaran para evaluar cada una de las características según los siguientes factores:

- UA: Umbral de adaptabilidad
- PO: Puntuación obtenida

Punto de escala genérica	Definición de punto de escala	Punto de Escala
Ninguno	Causa confusión. La forma en que se implementó la característica hace que sea difícil de utilizar y / o alentar el uso incorrecto de la característica	1
Muy bajo	No lo reconoce. La característica no se admite ni se refiere el manual de usuario	2
Bajo	La característica se admite indirectamente, por ejemplo mediante el uso de otras características de la herramienta en combinaciones no estándar.	3
Suficiente	La característica aparece explícitamente en la lista de características de las herramientas y el manual de usuario. Sin embargo, algunos aspectos de la característica de uso no son atendidos.	4
Alto	La característica aparece explícitamente en la lista de características de las herramientas y manual de usuario. Todos los aspectos de la característica están cubiertos, pero el uso de la característica depende de la experiencia del usuario	5
Muy alto	La característica aparece explícitamente en la lista de características de las herramientas y manual de usuario. Todos los aspectos de la característica están cubiertos y la herramienta proporciona cuadros de diálogo a la medida para ayudar al usuario.	6
Soporte total	La característica aparece explícitamente en la lista de características de las herramientas y manual de usuario. Todos los aspectos de la característica están cubiertos y la herramienta proporciona escenarios de usuario para ayudar al usuario como "Asistentes".	7

Tabla 3-5 Escala de juicio para evaluar el soporte de las características según DESMET

De esta manera se evaluarán cada una de las características para cada técnica/notación de análisis y modelado de tarea.

A continuación se procede a evaluar los criterios mencionados asociados a cada técnica/notación que hemos abordado con anterioridad teniendo en cuenta que se hará de forma parcial y total, esto es la parte parcial será evaluar a cada grupo (IPO e Ing. Software) y la parte total será la combinación de cada uno de los dos grupos como veremos más adelante.

Evaluación de técnicas basadas en UML

Las técnicas basadas en UML vienen de la disciplina de Ingeniería del software y se han demostrado útiles, en general, para modelar distintas vistas de un sistema informático. Entre esas vistas también están las tareas que éste ofrece.

La Tabla 3-6 muestra la evaluación, a través de DESMET, de las técnicas basadas en UML.

Criterio	IMP	UA	PO	DIF	RES
Int	3	7	5	-2	-6
Esc	3	7	4	-3	-9
Est	3	6	4	-2	-6
Sim	3	6	3	-3	-9
Not	3	6	4	-2	-6
Reu	3	5	5	0	0
Exp	2	7	4	-3	-6
Alm	1	5	6	1	1
Awa	4	5	3	-2	-8
Fee	4	4	4	0	0
Eva	3	5	4	-1	-3
Cal	2	3	4	1	2
Doc	3	4	6	3	9
Coo	3	3	4	1	3
Total					-38

Tabla 3-6 Evaluación para técnicas basadas en UML

De los resultados obtenidos se revela, en la Tabla 3-6, que esta técnica no es muy recomendable para tratar de cubrir criterios como el de 'Int'. Una de las justificaciones de este resultado podría ser que debido a que las técnicas basadas en UML se enfocan explícitamente en el proceso y no en el usuario, es decir, no toman en cuenta como el usuario interactúa con las aplicaciones desarrolladas, ni tampoco evalúa el uso que este le pudiera dar al sistema, al fin y al cabo, la aplicación puede funcionar de la forma en la que fue soñada a nivel de ejecución pero si los usuarios no se sienten cómodos o bien no entienden cómo usar la aplicación de nada servirá y se incurriría en mucho tiempo y dinero intentando que el usuario se adapte al uso que tiene en ese momento el sistema, pero al final siempre se sentirá incomodo ya que no satisface sus necesidades desde el punto de vista de su uso.

Otro de los puntos en los que las notaciones basadas en UML no parecen demostrarse adecuadas es en el de 'Esc' ya que con UML se crean diagramas para especificar diferentes perspectivas del sistema, p.e los diagramas de casos de usos describen los pasos para realizar un proceso dado y los de actividad representan los flujos de esos procesos, y mantener o tratar de retener esos diagramas es un poco tediosos por parte de los involucrados en el modelado de las aplicaciones.

Además, el criterio de 'Sim' no está cubierto del todo por las herramientas de modelado de UML, ya que no puede especificar diferentes tipos de tareas solamente las tareas que puedan

ser realizadas por la máquina que están directamente relacionadas con el proceso no así otros tipos de tareas realizadas por los usuarios p.e de tipo colaborativas; si bien es cierto que se pueden representar ideas complejas a través de estereotipos en UML indicando un significado adicional no es del todo suficiente ya que solamente quedaría plasmada como una idea abstracta y de nuevo se tendría que crear modelos a partir de esta idea. El awarenes tampoco se soporta directamente, ya que este criterio es exclusivo de los entornos colaborativos y al no poder describir tareas de tipo colaborativas es lógico que no pueda describir elementos que ayuden a enriquecer el awarenes una de las principales características presentes hoy en día en entornos ubicuos, además si se hiciera uso de estereotipos surge de nuevo el problema a en que la implementación de estos es un poco abstracto sugiriendo una idea general y no específica de lo que se desea representar.

Evaluación de HTA

La Tabla 3-7 muestra los resultados obtenidos al evaluar esta notación con las características descritas anteriormente.

Criterio	IMP	UA	PO	DIF	RES
Int	3	7	3	-4	-12
Esc	3	7	3	-4	-12
Est	3	6	2	-4	-12
Sim	3	6	2	-4	-12
Not	3	6	3	-3	-9
Reu	3	5	5	0	0
Exp	2	7	3	-4	-8
Alm	1	5	2	-3	-3
Awa	4	5	3	-2	-8
Fee	4	4	3	-1	-4
Eva	3	5	4	-1	-3
Cal	2	3	4	1	2
Doc	3	4	4	0	0
Coo	3	3	2	-1	-3
Total					-84

Tabla 3-7 Evaluación de HTA

La Tabla 3-7 muestra que esta notación no es mejor que las técnicas basadas en UML (ver Tabla 3-6), sin embargo entre los puntos débiles identificados por los expertos están que no representa muy bien el criterio de 'Int' asociado al alto grado de variabilidad de los sistemas lo que lo hace un punto muy importante ya que es muy común encontrar estas características en entornos colaborativos como los CSCW y los CVEs.

Esta notación no describe con precisión el criterio de 'Int' porque HTA se enfoca principalmente en aspectos cognitivos que tienen que ver con el proceso, es decir el conocimiento que tienen los usuarios de ciertas actividades o tareas que se llevan a cabo en

algún proceso de producción industrial dado no así en aspectos como el uso que los usuarios le darán al sistema.

De igual manera la 'Esc' es una de las debilidades más notorias que tiene esta notación ya que de la forma en cómo se estructuran los modelos en HTA, a través de cajas de textos o tablas, tiende a ser incomodo, tedioso y muy poco práctico por la forma rudimentaria en la que se define este tipo de modelos por lo que es uno de los criterios en los que falla. Al mismo nivel de fallo esta la simulación ya que al igual que las técnicas basadas en UML, HTA no hace diferencia de ningún tipo de tareas que deben de ser ejecutadas, esto es, en esta notación no se encuentra separada de forma explícita los tipos de tareas que se deben de producir el modelado de aplicaciones software ya que se enfoca más en los procesos que los usuarios realizan pero no en diferenciar que tipos de tareas podrían realizar los usuarios, las máquinas y a su vez que estos tipos de tareas en que entornos como la colaboración, la coordinación o la comunicación en CSCW.

Otro aspecto que no cubre esta notación es especificaciones ya realizadas a través de otras notaciones que se puedan de cierta forma complementar con este modelo, esto se debe a que es una notación cerrada ideada específicamente para que lo que se creen los modelos a partir de esta notación y en ningún momento se concibió que pudiese complementarse con otra ni mucho menos.

Luego el criterio de 'Exp' no es cubierta por esta notación ya que por su carácter HTA no toma en cuenta el uso que los usuarios le pudiesen dar a los sistemas desarrollados por lo que se ha dicho reiteradamente de que fue ideada para tomar en cuenta los procesos q los usuarios realizan no como estos manipulan las aplicaciones.

El 'Awa' es otra de las debilidades identificadas por el grupo de expertos ya que con esta característica en HTA no tiene algún mecanismo que nos permita representar de forma explícita como los usuarios usan el sistema, interactuando de forma coordinada en el mismo momento y en el mismo espacios o iguales o diferentes, con esta notación solamente se da una representación general de las tareas llevadas a cabo por los usuarios.

Evaluación de GOMS

La Tabla 3-8 muestra el resultado de evaluar esta notación con las características.

Criterio	IMP	UA	PO	DIF	RES
Int	3	7	3	-4	-12
Esc	3	7	3	-4	-12
Est	3	6	3	-3	-9
Sim	3	6	3	-3	-9
Not	3	6	3	-3	-9
Reu	3	5	2	-3	-9
Exp	2	7	3	-4	-8
Alm	1	5	3	-2	-2

Criterio	IMP	UA	PO	DIF	RES
Awa	4	5	3	-2	-8
Fee	4	4	3	-1	-4
Eva	3	5	3	-2	-6
Cal	2	3	2	-1	-2
Doc	3	4	2	-2	-6
Coo	3	3	2	-1	-3
Total					-99

Tabla 3-8 Evaluación de GOMS

La Tabla 3-8 muestra que GOMS es una de las notaciones mal situadas en la valoración del cuestionario por parte del grupo de expertos.

Dentro de los criterios más débiles que se le achaca a esta notación está en que no puede manejar sistemas altamente interactivos de forma eficiente lo que se traduciría en un escribir un 'programa' con muchas líneas de código.

En términos generales, en este estudio, esta notación no cubre aspectos como escalabilidad, aplicaciones altamente interactivas o la simulación. Además falla en otros criterios como la integración con la experiencia de los usuarios, la reutilización de especificaciones parciales y el awareness.

Evaluación de CTT

La Tabla 3-9 muestra la evaluación de esta notación por parte del grupo de expertos.

Criterio	IMP	UA	PO	DIF	RES
Int	3	7	4	-3	-9
Esc	3	7	2	-5	-15
Est	3	6	3	-3	-9
Sim	3	6	5	-1	-3
Not	3	6	2	-4	-12
Reu	3	5	4	-1	-3
Exp	2	7	3	-4	-8
Alm	1	5	3	-2	-2
Awa	4	5	2	-3	-12
Fee	4	4	3	-1	-4
Eva	3	5	4	-1	-3
Cal	2	3	4	1	2
Doc	3	4	4	0	0
Coo	3	3	4	1	3
Total					-75

Tabla 3-9 Evaluación de CTT

La Tabla 3-9 permite aseverar que CTT está limitada en algunos aspectos. El criterio de 'Esc' es uno de los puntos de los que adolece esta notación. Una de las razones podría ser que debido a la forma en cómo se estructura el modelado (a través de un árbol de tareas) lo hace un poco difícil de manejar y al tratar de modelar aplicaciones (p.e los juegos en el mundo del entretenimiento) complejas en la que se intente modelar todos los posibles usos que le den los usuarios al sistema se volvería algo difícil de entender y manipular por parte de los involucrados en el diseño y desarrollo del software.

Además, otra debilidad que presenta esta notación es que no es flexible al tratar de integrarla con otras notaciones existentes ('Not') y al igual que HTA en esta notación lo que se modela con ella es exclusivamente de ella y no se puede asociar a otro tipo de notación existente.

Luego está el awareness ('Awa'), si bien es cierto que con CTT se pueden definir tipos de tareas especiales como las colaborativas que están relacionadas con el awareness en los entornos colaborativos, CTT carece de poder de expresividad al tratar de enriquecer este tipo de elementos ya que solamente considera actividades que los usuarios realizan de forma conjunta, pero no considera que tipo de retroalimentación pudiese ser percibida por parte de estos para que estén al tanto de todo lo que pasa, p.e en el editor de documentos descrito anteriormente; mientras más conocimiento tenga un usuario lo que están haciendo los demás mucho mejor será la experiencia en el uso del sistema.

Otro de los puntos débiles de este tipo de notación está en la interactividad del sistema como hemos dicho anteriormente en el ejemplo de la industria del entretenimiento es difícil modelar con CTT todos los posibles escenarios en un ámbito tan complejo y variable puesto que al extender características a un nivel de detalle mayor los hace muy difícil de manejar, pero CTT tampoco puede manejar sistemas muy interactivos que está relacionado con la característica anterior.

3.3 Análisis global de los resultados

La Figura 3-6 muestra representación gráfica del análisis realizado con cada una de las técnicas/notaciones que hemos tratado. Los resultados muestran que ninguna de las técnicas cubre de forma satisfactoria las características que hemos descrito y que según nuestro criterio debería de tener una notación en Interacción Persona Ordenador.

Una vez que hemos visto que las notaciones existentes no cubren del todo las características que debería de presentar una técnica/notación ideal de tal manera que pueda ser integrada fácilmente a cualquier paradigma de interacción post-WIMP es necesario buscar la forma de cubrir esas características por lo que esto se realizara en el marco de las actividades de la propuesta de tesis doctoral de la que hablaremos en el próximo capítulo.

Luego, siguiendo con el uso de la metodología DESMET, se procede a realizar una comparativa del porcentaje que ha satisfecho cada característica en cada técnica/notación para tener una perspectiva general de lo que cubre cada técnica en cada característica.

La Figura 3-7 muestra el análisis basado en el porcentaje que hemos descrito.

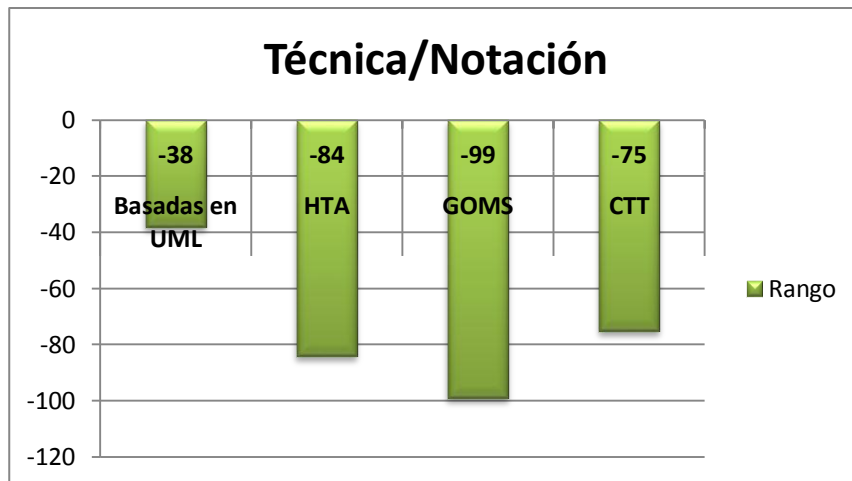


Figura 3-6 Resultado de evaluación de técnicas/notaciones

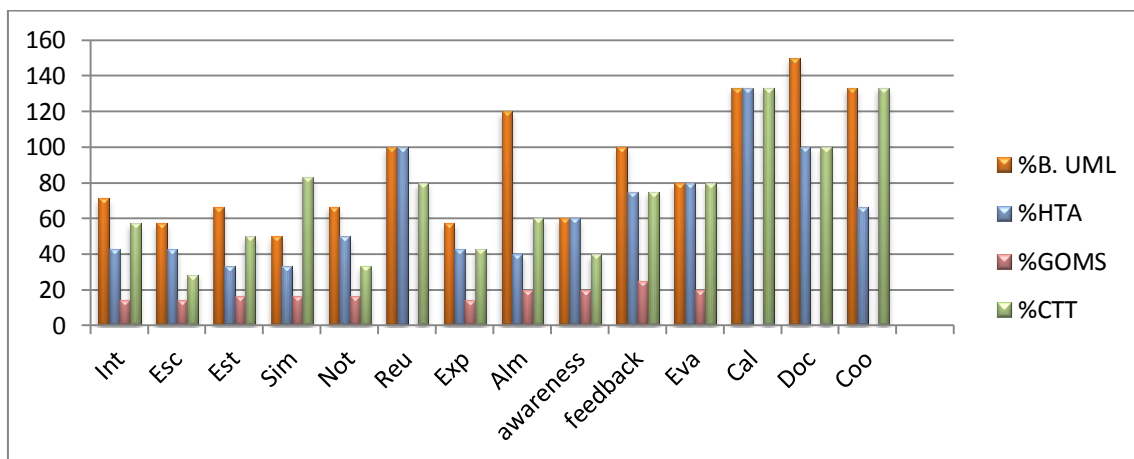


Figura 3-7 Resultado de comparativa en porcentaje de cada característica

Por lo que podemos observar de la Figura 3-7 la mejor aproximación para satisfacer las características nos lo dan las notaciones UML y CTT (esta última en especial) pero no de la forma en que se espera, si bien es cierto que cubren algunas características se quedan cortas en algunas otras que son de vital importancia al momento de diseñar interfaces de usuario independientes de la plataforma y la modalidad por lo que se requiere resolver esta problemática planteada.

3.4 Conclusiones

Las técnicas/notaciones de análisis y modelado de tareas son muy útiles en la interacción persona ordenador por lo que debido a las necesidades de cubrir los requerimientos que hoy día y en un futuro cercano pudiesen presentar es de mucha importancia ya que es la base de una interacción más próxima a un mundo más intuitivo para los usuarios por lo que enfocarnos en abarcar los criterios que hemos definido sería un gran paso en esta disciplina.

Los aportes que existen actualmente no cubren de forma satisfactoria estos criterios que presentan actualmente ya que a día de hoy vivimos en un mundo más interactivo y más cercano unos con otros en donde las tareas se realizan de forma coordinada y conjunta siendo la comunicación uno de los pilares que se tienen que tener en cuenta. De esta manera hemos

visto que algunos de estos aportes como CTT cubren de forma muy vaga estos criterios no de la forma en la que se requiere a un nivel muy alto, además técnicas basadas en UML si bien es cierto que cubren de forma excelente los procesos del sistema carecen de poder de expresividad con las tareas relacionadas a los usuarios y esto es algo lamentable porque al final el usuario es el encargado de juzgar el éxito o fracaso de un sistema software.

Además, hemos visto como HTA y GOMS son las notaciones que menos cubren las características presentadas quizás porque sean aportes que se presentaron hace muchísimos años y no tenían en cuenta la evolución que han tenido las interfaces que hoy en día se desarrollan y el nivel de complejidad en un mundo en el que los usuarios están cada vez más cerca virtualmente hablando.

Debido a que los resultados que hemos presentado y dada la importancia que presentan los criterios que hemos definido se realizarán más réplicas de los experimentos realizados a partir de los criterios que ya hemos definido. Las réplicas abarcarán un espacio muestral más amplio para despejar cualquier duda que pueda surgir ya que los resultados arrojados por estos experimentos no son del todo satisfactorios, el experimento consistirá en realizar nuevamente una evaluación de cada una de las notaciones abordadas.

De esta forma en el próximo capítulo propondremos la generalización de una forma de trabajar y evaluar las notaciones, metodologías y herramientas ligadas al desarrollo de interfaces de usuario. Dicha propuesta constituirá la Tesis Doctoral.

Capítulo 4

Anteproyecto de Tesis

4.1 Introducción

El desarrollo de interfaces basado y/o dirigido por modelos (Mb-UIDE/MDUI) es el aporte más aceptado en IPO para desarrollar interfaces de usuario. A partir de esta propuesta se han tratado de estandarizar ciertos mecanismos para que este proceso sea aceptado como único. Entre estas propuestas de desarrollo conducido por modelos surgió la propuesta de (Puerta 1997), que clasificaba los modelos en presentación, domino y tareas, y recientemente la propuesta de un marco de trabajo denominada CAMELEON (ver sección 2.3) que clasifica diferentes modelos ontológicos en modelos de tipo de dominio como conceptos y tareas, al contexto de su uso entre los que están el usuario la plataforma y el ambiente, y la adaptación.

El modelo de tareas, y por consiguiente las técnicas de modelado de tareas que ya hemos abordado en el capítulo anterior, son parte del marco de trabajo de CAMELEON para desarrollar interfaces de usuarios en los que se tenga en cuenta como elemento fundamental al usuario es por ello que han surgido una serie de aportaciones en los otros tipos de modelos como los de interfaces de usuario abstracta y específica (AUIs y CUIs) y el Final UI, que tratan de englobar diferentes niveles de desarrollo desde la etapa más temprana que es el análisis y hasta la última de esta que es la interfaz de usuario final.

Las actividades realizadas en el capítulo anterior han permitido establecer criterios de cuestionamiento y de resolución de conflictos con los que determinar si las notaciones disponibles y utilizadas actualmente en las propuestas Mb-UIDE, son las adecuadas para abordar los retos que impone el desarrollo de los futuros y previsibles nuevos tipos de interfaces de usuario y de posibilidades de interacción. Por ello, y a modo de anticipo, proponemos en este anteproyecto de tesis continuar con el análisis y discusión de los principales modelos asociados a las propuestas Mb-UIDE y considerar en dicho análisis las notaciones disponibles para su soporte.

4.2 Dominio del problema

En la actualidad es común encontrar en el desarrollo de interfaces con diferentes formas de interacción entre los usuarios a través de diferentes dispositivos y modalidades; desde móviles táctiles en la que los usuarios manipulan los objetos, usuarios interactuando a través de llamadas consultando información o respondiendo a encuestas, en el ocio a través de juegos de ordenador a ordenador tan populares, en la edición de documentos en línea a través de la colaboración entre los usuarios agrandando la complejidad ya que la edición la realizan de forma simultánea (o no) en el mismo espacio y tiempo o bien en diferentes ambos.

De esta manera los resultados de esta propuesta de tesis doctoral tiene un amplio dominio de aplicación tales como el investigativo, médico, comercial, educativo y estatal.

En fin, hay tantas formas de interacción y plataformas por lo que esta realidad nos sugiere definir de forma veraz y objetiva la mejor herramienta, propuesta o modelo teniendo en

cuenta el contexto en el que se realice la interacción. Es por ello que esta propuesta de tesis doctoral tiene como uno de sus objetivos presentar mecanismos que permitan elegir aportes que mejor se adapten a estos contextos, es decir, sugerir alternativas de selección en diferentes etapas del desarrollo de sistemas tales como AUIs, CIOs, Final UI.

4.3 Propuesta de tesis

El contribuir a mejorar el desarrollo de interfaces de usuario, que se propone en este anteproyecto de tesis doctoral, está relacionado directamente con establecer mecanismos de justificación y selección a través de un análisis profundo sobre las distintas propuestas que han surgido en Mb-UIDE, acrónimo que aglutina las propuestas más recientes y más aceptadas. (Ver Figura 4-1)

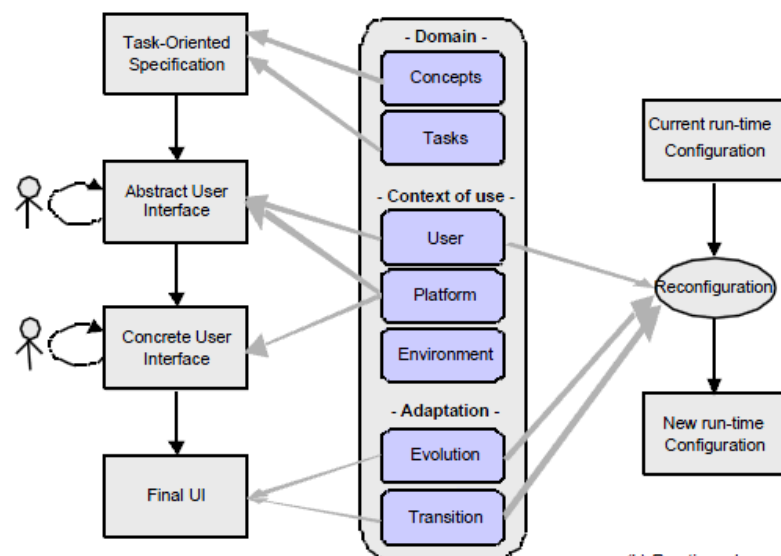


Figura 4-1 Modelos del Framework CAMELEON para fases de diseño y ejecución

De esta manera, definir mecanismos de selección sensibles al contexto, plataforma y modalidad, y por ende a las diferentes fases del desarrollo de interfaces, en el que se desarrolle la interacción es de vital importancia porque será una forma objetiva de elegir aportes basados en la mejor descripción o adaptación que estos pudieran tener.

Con la realización de esta propuesta de tesis doctoral se pretende dar respuesta a las siguientes hipótesis:

1. Primero, analizar de manera justificada y empírica si las técnicas y notaciones existentes que se utilizan en los diferentes modelos (tareas, usuarios, etc.) y niveles (AUIs, CUIs, etc.) de Mb-UIDE son realmente adecuadas, ya que como hemos visto en el capítulo anterior en el trabajo de investigación, p.e para el modelo de tareas las propuestas de técnicas/notaciones de análisis y modelados de tareas no han salido muy bien paradas, ya que no se adaptan satisfactoriamente a los criterios que hemos establecidos como de vital importancia en el desarrollo de interfaces de usuarios actuales y previsibles. Es por ello que trataremos de dar respuesta a este punto a través de un análisis profundo en los que se aborden diferentes modelos, niveles atendiendo a la propuesta hecha en CAMELEON en (Vanderdonckt J. 2002). Esta

hipótesis partirá de que las notaciones actualmente disponibles son susceptibles de ser utilizadas, pero que no todas ellas ofrecen las mismas facilidades.

2. Segundo, analizar si las notaciones visuales, tal como usiXML, que utilizan los modelos de interfaces de usuarios abstractas (AUIs) son adecuadas. Esto se realizará a través de la propuesta de (Sangiorgi U 2011) que nos permite evaluar diferentes aspectos de las notaciones visuales que se han propuesto para mejorar el diseño y desarrollo de interfaces de usuario al nivel de abstracción.
3. Tercero, analizar si las herramientas existentes de modelado de los distintos modelos asociados al desarrollo de interfaces de usuario y evaluar si dichas herramientas cumplen con los retos y requisitos que presenta actualmente la especificación de interfaces de usuario. En este momento, ligado al proceso de estandarización de algunas de las propuestas Mb-UIDE, distintos proyectos europeos han propuesto el desarrollo de distintos editores y herramientas destinados a facilitar la especificación de distintos modelos asociados a interfaces de usuario. En este sentido, un resultado previsible de esta tarea será el establecimiento de criterios que permitan evaluar, comparar y seleccionar la herramienta más adecuada.
4. Cuarto, saber si las notaciones empleadas en el modelado de interfaces de usuario son independientes de la plataforma y modalidad de interacción tales como CVEs, realidad virtual, aumentada y diferentes formas de interactuar como GUI, VUIs, etc. con la misma técnica de análisis y modelado de interfaces de usuario para cada plataforma y/o modalidad. Así, por ejemplo, a través de casos de estudio se propone modelar las distintas vistas ofrecidas por la interacción que realizan los usuarios
5. Quinto, cuestionarnos si el uso conjunto de las notaciones seleccionadas es significativa. Tal y como hemos hecho en el trabajo de investigación para los criterios, con la particularidad de que aún falta realizar experimentos y sus réplicas a través del modelado de algún problema en cada paradigma en los que sé que se aborden las técnicas de modelado de tareas y sus notaciones para poder obtener diferencias y/o similitudes para responder a este punto.
6. Sexto, las notaciones propuestas para el modelado de tareas son ortogonales. Aquí se analizará si las técnicas de análisis y modelado de interfaces de usuario tienen puntos en común y que traten de generalizar actividades de los usuarios en cualquier entorno o paradigma de interacción.
7. Puesta en práctica de las conclusiones alcanzadas. Una vez determinadas las técnicas y notaciones mejor situadas para dar soporte al desarrollo y especificación de interfaces de usuario se tratará de poner en práctica dichas técnicas y notaciones y evaluar cuantitativa y cualitativamente los resultados obtenidos de su utilización. Para ello se recurrirá a la selección de un conjunto de escenarios significativos.

De este modo, esta tesis contribuirá a estudiar técnicas de especificación de requisitos que se adaptan a la especificación de los distintos niveles de abstracción propuestos en CAMELEON. El desarrollo de interfaces de usuario dirigido y basado en modelos requiere que el trabajo realizado en la especificación sea capturado en modelos que permitan la automatización de tareas tales como la transformación de modelos (ver Figura 4-2).

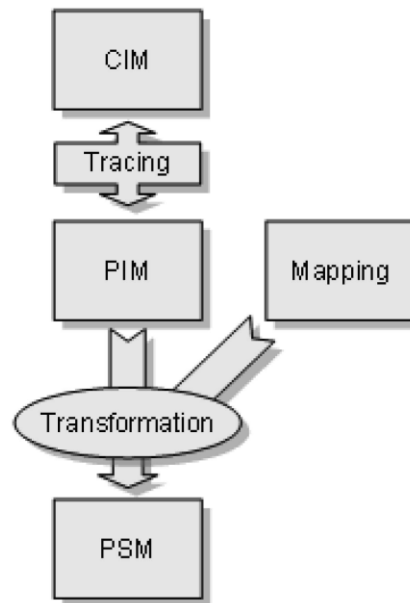


Figura 4-2 Modelos en MDD

Una vez que hemos definido las hipótesis de la propuesta de tesis doctoral se procederá seguidamente a presentar la planificación para llevar a cabo cada uno de los puntos listados anteriormente.

4.4 Planificación de actividades

Además del análisis y elección justificada de las técnicas y notaciones mejor situadas para dar soporte a la especificación y desarrollo de interfaces de usuario, en este anteproyecto de tesis doctoral se aportará un resultado adicional que será la definición de un marco de trabajo o framework para abordar las actividades de análisis llevadas a cabo. Bajo un prisma Investigación-Acción en las actividades desarrolladas en la tesis doctoral que se propone se identifican los siguientes involucrados:

- El *grupo de investigación* que realizara el estudio. El rol será realizado por el doctorando, D. Jaslin Fernando Flores, y contará con la colaboración de los integrantes del grupo de investigación LoUISE de la Universidad de Castilla-La Mancha (UCLM).
- El *objeto de estudio* o problema que se debe resolver. El problema a resolver será el de realizar un análisis profundo y justificado, asociando un marco de trabajo que guíe las actividades de dicho análisis, sobre las distintas técnicas y notaciones asociadas a las propuestas Mb-UIDE.
- El *equipo de referencia* recibirá el resultado del estudio y participará de forma activa en el proceso de investigación. El rol será llevado a cabo por grupos de expertos en cada una de los paradigmas de interacción de tal manera que ellos puedan validar los resultados obtenidos en el estudio. También se considera involucrar a distintas empresas dedicadas al desarrollo del software, que podrán utilizar y ofrecer escenarios reales relacionados con el desarrollo de interfaces de usuario.
- El *beneficiario del estudio*, es decir quién o quienes harán uso del estudio. El rol será llevado a cabo por cualquier institución pública o privada dedicada o involucrada en

actividades de desarrollo de interfaces de usuario utilizando técnicas basadas en modelos.

Una vez que se han definido los actores de la investigación, estos mismos estarán inmersos en un conjunto de actividades y tareas que definimos a continuación (ver Figura 4-3):

Actividad 1. Revisión y estudio del estado del arte relacionado con la investigación.

- Tarea 1. Estudio y familiarización con las propuestas de Desarrollo basada en Modelos (Mb-UIDE). Estudio del marco de trabajo CAMELEON.
- Tarea 2. Estudio y familiarización de los lenguajes de especificación de interfaces de usuario. Atención especial a la propuesta usiXML, actualmente en proceso de estandarización.
- Tarea 3. Estudio y familiarización de las propuestas destinadas a la evaluación y análisis de metodologías, procesos y herramientas disponibles en la literatura. Así como identificar las ventajas y limitaciones de la realización de técnicas basadas en cuestionarios. Elaboración, en su caso, de una propuesta de evaluación destinada a dar soporte a un marco de evaluación de notaciones, técnicas y herramientas ligadas al desarrollo de interfaces de usuario.
- Tarea 4. Identificación de herramientas software que den soporte a las anteriores tareas recogidas en la Actividad 1.

Actividad 2. Estudio, evaluación y discusión de notaciones destinadas a dar soporte al desarrollo de interfaces de usuario.

- Tarea 1. Identificación de criterios significativos de selección y elaboración de los modelos básicos y extendidos asociados a la especificación de interfaces de usuario.
- Tarea 2. Evaluación cuantitativa y cualitativa de los modelos asociados al desarrollo de interfaces de usuario.
- Tarea 3. Evaluación cuantitativa y cualitativa de las notaciones asociadas a la especificación de interfaces de usuario.
- Tarea 4. Recopilación de buenas prácticas, fortalezas, debilidades y oportunidades asociadas a la especificación de interfaces de usuario, en función de los resultados obtenidos con anterioridad.

Actividad 3. Estudio, evaluación y discusión de técnicas, metodologías y procesos destinados a dar soporte al desarrollo de interfaces de usuario.

- Tarea 1. Identificación de criterios significativos asociados a las metodologías y procesos asociados al desarrollo de interfaces de usuario.
- Tarea 2. Evaluación cuantitativa y cualitativa de metodologías y procesos destinados a soportar el desarrollo de interfaces de usuario utilizando técnicas basadas en modelos.

- Tarea 3. Recopilación de aspectos comunes y diferentes entre las distintas propuestas metodológicas, que bajo el punto de vista Mb-UIDE, permiten el desarrollo de interfaces de usuario.

Actividad 4. Estudio, evaluación y discusión de herramientas destinadas a dar soporte al desarrollo de interfaces de usuario.

- Tarea 1. Identificación de requisitos, retos, desafíos y carencias asociados a las herramientas que dan soporte al desarrollo de interfaces de usuario bajo la filosofía Mb-UIDE.
- Tarea 2. Evaluación cuantitativa y cualitativa de herramientas destinadas a soportar el desarrollo y especificación de interfaces de usuario.
- Tarea 3. Identificación, análisis crítico y documentación de las limitaciones y posibilidades ofrecidas por las herramientas de soporte a la elaboración de interfaces de usuario.

Actividad 5. Documentación, diseminación y difusión de actividades.

- Tarea 1. Documentación y elaboración de la memoria asociada a la Tesis Doctoral.
- Tarea 2. Selección de congresos, conferencias y publicaciones relevantes y asociadas a los temas tratados en la Tesis Doctoral y confección de artículos destinados a esos elementos de difusión de actividades de investigación.

4.5 Cronograma

Seguidamente, en la Figura 4-3, se representa gráficamente la secuenciación y concurrencia entre las actividades y tareas identificadas con anterioridad.

Cabe destacar que las actividades y tareas identificadas se llevaran a cabo de forma iterativa e incremental para llevar el proceso de investigación de forma eficiente.

Actividad	Tarea	Año 1	Año 2	Año 3
Estado del arte				
	Estudio de CAMELEON			
	Estudio de usiXML			
	Análisis de metodologías, procesos			
Evaluación y discusión de notaciones	Identificación de herramientas software			
	Identificación de criterios			
	Evaluación cuantitativa y cualitativa de Mb-UIDE			
	Evaluación cuantitativa y cualitativa de las notaciones			

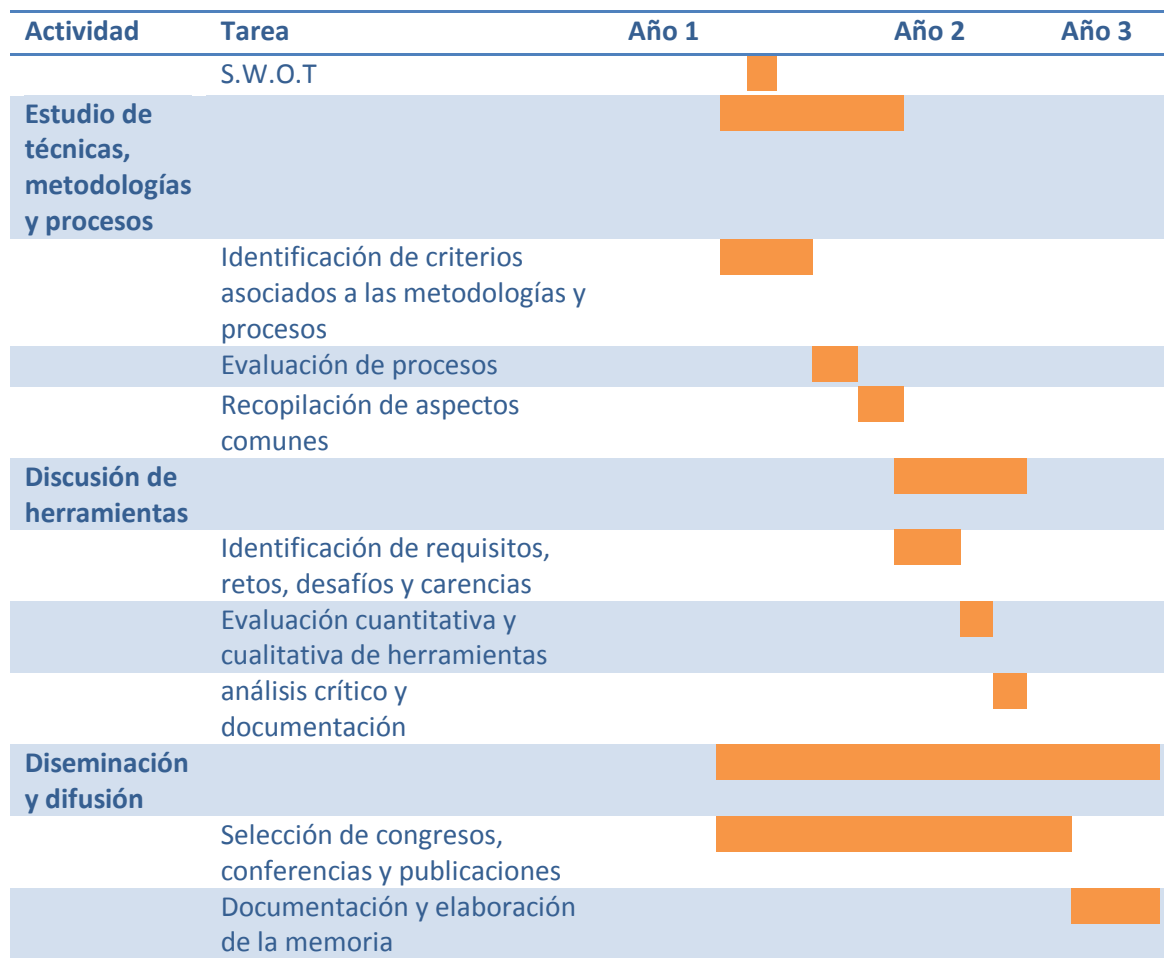


Figura 4-3 Cronograma de actividades

Capítulo 5

Currículo Vitae

5.1 Titulación académica

Ingeniería en sistemas

Universidad Nacional de Ingeniería

Febrero 2008

5.2 Becas Disfrutadas

Universidad Nacional de Ingeniería

Beca a la excelencia académica

01/03/2003-20/12/2007

Facultad de Ciencias y Sistemas, Universidad Nacional de Ingeniería

Beca de colaboración: Desarrollo de aplicaciones cliente/servidor con JSP y MySQL

03/03/2004-03/09/2004

Facultad de Ciencias y Sistemas, Universidad nacional de Ingeniería

Beca de colaboración: Desarrollo de aplicaciones web con PHP, ASP .NET y SQL Server

15/10/2004-03/04/2005

Facultad de Ciencias y Sistemas, Universidad nacional de Ingeniería

Beca de colaboración: Administración de Redes y Servidores

01/05/2005-30/11/2006

Fundación Carolina

Beca de investigación: Análisis y Modelado de tareas en los nuevos paradigmas de interacción hombre-máquina

03/10/2011-16/07/2012

5.3 Otros Méritos

5.3.1 Experiencia profesional

Responsable laboratorio de ordenadores de la Facultad de Ciencias y Sistemas en la Universidad Nacional de Ingeniería

De Enero 2005 a Diciembre 2006.

Desarrollo de aplicaciones web en Grupo Consultores Sociales (GCS) en plataformas Java Server Page (JSP) y Servlet con Microsoft SQL Server, MySQL y ORACLE

De Enero 2007 a Noviembre 2007.

Desarrollador Junior de aplicaciones, TELMARK Nicaragua, S.A

De Enero 2008 a Diciembre 2008

Analista-Programador de sistemas de información, ENITEL-CLARO América Móvil

De Febrero 2009 a Noviembre 2009

Consultor de Sistemas de tecnologías de información y comunicación en el Ministerio de Educación Cultura y Deportes

De Noviembre 2009 a Febrero 2011.

Consultor informático en el Instituto Nicaragüense de Seguridad Social

De Marzo 2011 a Septiembre 2011.

5.3.2 Otros cursos

Universidad nacional de Ingeniería

Curso de Gerencia de Recursos Humanos

01/09/2009-15/02/2010

Universidad nacional de Ingeniería

Curso de Desarrollo Organizacional

25/02/2010-01/06/2010

Universidad nacional de Ingeniería

Curso de Marketing

15/06/2010-01/10/2010

Universidad nacional de Ingeniería

Curso de Alta Gerencia

15/10/2010-10/02/2011

Referencias

- Annett J. & Duncan, K. "Task analysis and training design." *Occupational Psychology*, 1967: 211–227.
- Baskerville R., Pries-Heje J. "Grounded action research: a method for understanding IT in practice." *Accounting, Management and Information Technologies*, 1999: 1-23.
- BLACS. 30 06, 2012. <http://www.netlib.org/blacs/>.
- BLAS (*Basic Linear Algebra Subprograms*). 2012. <http://www.netlib.org/blas/>.
- Caffiau, S. "Increasing the expressive power of task analysis: Systematic comparison and empirical assessment of tool-supported task models." *Interacting with Computers*, 2010: 569-593.
- Chuck, L. *Hadoop in Action*. Manning Publications Co., 2011.
- da Silva, P. "UMLi: the unified modeling language for interactive applications." *Proceeding UML'00 Proceedings of the 3rd international conference on The unified modeling language: advancing the standard*. Berlin: Springer-Verlag Berlin, Heidelberg ©2000, 2000. 117-132.
- Diaper D., Annett, J., and Stanton, N.A. *The Handbook of Task Analysis for Human-Computer Interaction*. Mahwah, NJ: Lawrence Erlbaum Associates, 2004.
- García, A. *Estudio y mejoras de la interacción en entornos virtuales colaborativos*. Albacete, 2010.
- Kieras, D. *A guide to GOMS task analysis*. Michigan, 1994.
- Kitchenham, B. *DESMET: A method for evaluating Software Engineering methods and tools*. Staffordshire, 1996.
- LaTeX. 05 20, 2012. <http://es.wikipedia.org/wiki/LaTeX>.
- Limbourg, Q. "Comparing Task Models for User Interface Design." In *The handbook of task analysis for Human-Computer Interaction*, by D Diaper, 135-155. London: LAWRENCE ERLBAUM ASSOCIATES, PUBLISHERS, 2003.
- López Sánchez M, Muñoz Márquez M, Rodríguez Chía A. *Inferencia Estadística*. Cádiz: Servicio de Publicaciones de la Universidad de Cádiz, 2006.
- Lopez-Jaquero, V. *Interfaces de usuarios adaptativas basadas en modelos y agente software*. Albacete, 2005.
- Lozano, M. *Entorno metodológico orientado a objetos para la especificación y desarrollo de interfaces de usuario*. Albacete, 2002.

- Luna, A, and G Moreno. "Fuzzy Logic Programming for Implementing a Flexible XPath-based Query Language." *Journal Electronic Notes in Theoretical Computer Science*, 2012: 3-18.
- M, López Sánchez, Muñoz Márquez M, and Rodríguez Chía. *Estadística Descriptiva y Probabilidad*. Cádiz: Servicio de Publicaciones de la Universidad de Cádiz, 2006.
- Markopoulos, P. "Modelling user task with the Unified Modelling Language." *IPO, Center for User-System Interaction, Eindhoven University of Technology*, 2002: 1-6.
- Mattsson A, Lundell B, Lings B, Fitzgerald B. "Linking Model-Driven Development and Software Architecture: A Case Study." *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING* (IEEE TRANSACTIONS ON SOFTWARE ENGINEERING) 35, no. 1 (2009): 1-11.
- Molina, J. *Un enfoque estructural para el desarrollo de interfaces 3D*. Albacete, 2007.
- Montero, F. *Integración de calidad y experiencia en el desarrollo de interfaces de usuario dirigido por modelos*. Albacete, 2005.
- Montero, Fr. *Google maps y sus posibilidades*. Mc Graw-Hill, 2010.
- MPI Documents. 06 30, 2012. <http://www.mpi-forum.org/docs/>.
- Paternò, F. *Model-based design and evaluation of interactive applications*. Springer, 2000.
- PBLAS Home Page. 04 03, 2012. http://www.netlib.org/scalapack/pblas_qref.html.
- Pribeanu, C. "Challenges for a task modeling tool supporting a task-based approach to user interface design." 2011: 1-4.
- Puerta, AR. *A model-based interface development environment*. 1997. <http://www.arpuerta.com/pdf/ieee97.pdf> (accessed 02 11, 1997).
- Ruiz, V. *modelo de Proceso para el Desarrollo de Interfaces en Entornos CSCW Centrado en los Usuarios y Dirigido por Tareas*. Albacete, 2007.
- Rumbaugh, J. *OMG Unified Modeling Language Specification*. 2000. <http://www.omg.org/spec/UML/> (accessed 03 01, 2000).
- Sangiorg U, Tesoriero R, Beuven F, Vanderdonck J. "Inspecting Visual Notations for USIXML's Abstract UI and Task models." *Proceedings of UIDL'2011 - Software Support for User Interface Description Language - Interact'2011*, 2011: 1-7.
- Schlunbaum, E. *Model-based User Interface Software Tools Current state of declarative models*. 11 1996. <http://smartech.gatech.edu/bitstream/1853/3516/1/96-30.pdf> (accessed 03 10, 2012).
- The R Project for Statistical Computing*. 2012. <http://www.r-project.org/> (accessed 11 01, 2011).

- Vanderdonckt J., Calvary G., Coutaz J., Thevenin D. *The CAMELEON Reference Framework*. Paris, 2002.
- Vazquez, C, and Moreno G. *Multi-Adjoint Lattices in Practice using FLOPER*. 2011. <http://dectau.uclm.es/floper/> (accessed 06 01, 2012).
- Weka 3 - Data Mining with Open Source Machine Learning Software in Java*. 2012. <http://www.cs.waikato.ac.nz/ml/weka/> (accessed 06 20, 2012).