

Evaluación realista de la usabilidad en la Ingeniería de la Interfaz Persona Ordenador

Juan Falgueras
Depto. de Lenguajes
y Ciencias de la Computación
Universidad de Málaga,
Campus de Teatinos s/n
juanfc@lcc.uma.es

Antonio Guevara
Depto. de Lenguajes
y Ciencias de la Computación
Universidad de Málaga,
Campus de Teatinos s/n
guevara@lcc.uma.es

Resumen

En este trabajo se muestra cómo las pretendidas técnicas de evaluación empíricas deben ser complementadas con técnicas de evaluación *a priori* basadas en el análisis léxico-semántico de los algoritmos fuente de las interfaces persona ordenador. Para ello se justifica una hipótesis universal basada en anteriores trabajos válidos en la medición de la complejidad de grandes programas de software debida a McCabe, aplicándola al proceso cognitivo de la interacción como proceso de complejidad similar al del propio algoritmo pero sustentado, no ya en el número de decisiones-bucles sino en el número de componentes activos que ofrece la interfaz. Así, una vez aceptada esta hipótesis, se plantean dos fenómenos interactivos a evaluar: la complejidad estática y la complejidad dinámica. La primera correspondería a la dificultad de reconocimiento y de decisión reunida en cada diálogo modal con el usuario. La segunda, la cantidad de decisiones entre las alternativas a las que el usuario se ve sometido durante el proceso de interacción, en el tiempo. Una vez establecidos, por una lado una hipótesis de trabajo para el análisis del código fuente y, por otro, elementos medibles, se establecen técnicas realistas de medición basadas en parsers directos sobre el código fuente de los programas durante el desarrollo, que darán medidas absolutas o relativas de la complejidad de los diálogos que se van programando y, por otro, una técnica de sondado y seguimiento de la interacción en el tiempo del usuario con el programa. Dada la importancia y complicación añadida del estudio de los sistemas cooperativos, esta última técnica adquiere especial relevancia en ellos, permitiendo llegar a construir sistemas cooperativos con interfaces razonablemente adaptativas a los perfiles de los usuarios conforme los usan y sin los costosos y lentos mecanismos de evaluación empíricos clásicos. El resultado es una batería de herramientas basadas en una revolucionaria metodología de evaluación de la usabilidad que, en última instancia, orientará al desarrollador durante el proceso de programación.

Palabras clave: Usabilidad, Evaluación formal, Sistemas Cooperativos

1 Introducción

Con [14], simplificando la definición de proceso interactivo como aquél *dirigido por el usuario*, estaríamos suponiendo que tenemos usuarios li-

bres y sistemas absolutamente flexibles. La situación real es más bien *la opuesta*, el proceso *dirige* más al usuario que el usuario al proceso. Ciertamente se espera que el usuario identifique metáforas, ya bien en forma creativa ge-

nerando así nuevo conocimiento (por la creación de *similaridad*, [5]), que puede ser más o menos afortunada para guiar intuitivamente al usuario, [15]. Estas metáforas, por la propia naturaleza de los sistemas físicos usados en los ordenadores son, sobre todo, gráficas o basadas en sonidos. Este conocimiento adquirido o la intuición previa del usuario reclamada por la interfaz para el reconocimiento de las metáforas empleadas ayudaran al usuario a “dirigir” (más bien a “dirigirse entre”) las siguientes acciones. En diversos trabajos anteriores, pero especialmente en [4], el autor ha realizado una extensiva labor de recopilación de las diversas técnicas actualmente en vigor para la evaluación de la usabilidad de interfaces de usuario concretos. En esta presentación el autor intentará mostrar la importancia de taxonomizar los *elementos interactivos* de las interfaces para, de esa forma, a partir de las clases formadas, tener grupos simples, fáciles de reconocer, suficientemente abstractos que permita la evaluación comparativa de términos típicos en la medición de la usabilidad.

Precisamente uno de los problemas más graves a los que se enfrenta el estudio científico de la Interfaz Persona Ordenador es el de la ambigüedad de los objetivos perseguidos. La única solución a esta cuestión la ofrecen las medidas prácticas de los resultados del uso de las aplicaciones.

En este sentido la mayoría de los métodos de usabilidad en ingeniería podrían contribuir de manera sustancial a la usabilidad con el mero hecho de llegar a ser usados durante el ciclo de vida del proyecto. Pero aún el mejor de los métodos es inútil si no se llega a usar nunca. Desafortunadamente, la experiencia muestra [8, 9] que la mayoría de los desarrolladores encuentran demasiado caros, intimidantes, difíciles y pesados de usar los métodos de usabilidad. Incluso métodos como el de evaluación heurística, [10], o la evaluación heurística económica, [9] (que utiliza de tres a cinco evaluadores en sucintas sesiones programadas) no siempre llegan a usarse.

Por otro lado, es altamente deseable como apoyo a los métodos ortodoxos de ingeniería de la usabilidad que el diseñador dispusiese de alguna **herramienta** que le permitiera hacer una estimación de la complejidad previa al desarrollo y que se está introduciendo en el diseño de

la Interfaz Persona Ordenador. Sin estos datos el analista y el diseñador han de esperar a una evaluación del prototipo para modificar su trabajo.

Los métodos formales útiles en las técnicas predictivas (como GOMS, TAG y derivadas), aún apoyándose en las características abstractas que se podrían extraer de un análisis formal de la interfaz, no pueden llegar a concretar una medida práctica, realista, de la compleja situación tanto ambiental, en la que se usa el sistema, como cognitiva del comportamiento humano. Esto es especialmente cierto en sistemas **cooperativos**, donde el comportamiento no se limita al de un usuario en un entorno sino que hay que coordinar las acciones de multitud de usuarios orientadas a diversidad de tareas [13] y tener en cuenta aún aspectos sociales.

No queremos ser redundantes sobre la consabida importancia actual de la Interfaz Persona Ordenador conforme se expande explosivamente su alcance a todo tipo de usuarios. Nos queremos centrar en aspectos eminentemente prácticos que lleven a resultados de calidad cercanos al sentido común. En este sentido es necesario situarse no en el plano teórico tratando de analizar el comportamiento del ser humano ni hacer un análisis formal de todos los algoritmos involucrados en una aplicación interactiva. Nos situaremos en los aspectos reales del desarrollo ingenieril de un sistema interactivo.

En este trabajo, aún en desarrollo, vamos a presentar una relación entre el proceso ingenieril de análisis, diseño y codificación de la interfaz con el resultado en términos de usabilidad. Como veremos en el siguiente apartado, las técnicas experimentales no están dando los resultados previstos, mientras que el desarrollo del software empieza a entrar en etapas de una madurez suficiente como para ofrecer valiosos parámetros aprovechables en la evaluación de la usabilidad final de los productos perseguidos. Particularmente nos centraremos en resultados presentados en [4], para aún alejándonos más de aquél planteamiento concretado en lenguajes genéricos, examinar las universalidad de las hipótesis presentadas allí.

2 Técnicas experimentales de evaluación

Los métodos y técnicas experimentales tratan con datos observados en el mundo de usuarios reales en el desarrollo de tareas con artefactos físicos. Estos artefactos incluyen escenarios sobre el papel, caricaturas, prototipos computacionales o plataformas tipo “Mago de Oz” (WOz). Las técnicas WOz han sido aplicadas tan sólo mediante interfaces en lenguaje natural, pero son especialmente representativas del tipo de evaluación que estamos hablando. Los datos relativos al *comportamiento del usuario* se graban en cinta y después se recogen e interpretan subjetivamente. Esta tarea requerirá tiempo y experiencia. Estos métodos son fundamentales para cualquier estudio **real** de la interacción ya que aportan parámetros específicos que después se utilizan en los estudios teóricos generalizando y extrapolándolos. Son, pues, el laboratorio más implacable de cualquier teoría sobre la interacción.

Existen herramientas de software para el seguimiento (monitorizado) de las actividades de usuarios, alumnos en aulas de informática, *beta-testers*, etc. que pueden permitirnos observar a la persona en los detalles más significativos básicos de la interacción, como movimientos de ratón o pulsaciones de teclado.

Nuestro objetivo se centra en la sustitución de tales herramientas por otras integradas en el propio software en desarrollo de manera que lo convierta en aplicaciones auto-evaluables.

Un aspecto muy importante a cubrir en este tipo de herramientas experimentales de evaluación es la **multimodalidad** de las posibles entradas del usuario. Nos referimos aquí a los posibles modos de interactuar que el sistema ofrezca al usuario: ratón, teclado, voz, imagen, etc. Esta diversidad es difícil de recoger y mantener con un sistema de observación humano como el clásico WOz. Si se trata además de realizar **estadísticas** de determinadas secuencias de acciones, para establecer frecuencias de uso, y posibles, por tanto **optimizaciones parciales**, es imprescindible la recogida automática de datos, difícil con sistemas no conectados a los procesos interactivos.

3 Sistemas actuales mixtos

En [1, 3] se presenta un sistema de evaluación semiautomatizada capaz de recoger datos recibidos en forma multimodal. Esta evaluación automática está basada en cuatro pasos:

1. definición del modelo de tareas. Particularmente siguiendo modelos análogos a los modelos GOMS, [2, 6], llegaríamos a un modelo conceptual de tareas (las hojas del árbol jerárquico del tareas, conceptualmente indivisibles).
2. adquisición de los datos conductuales del usuario. Esta recogida incluye la grabación de imágenes del usuario para la posterior interpretación de expresiones y gestos.
3. identificación de patrones de comportamiento. La repetición de acciones dentro de una tarea, debida a la forma de implementación de la interfaz. Por ejemplo, el reajuste sistemático del tamaño de las ventanas cada vez que se abren puede indicar un mal diseño de esta función de apertura, que no adecúa correctamente la presentación de las ventanas a las necesidades del usuario.
4. crítica de los mismos. Bajo una taxonomía debida a [12]: compatibilidad, homogeneidad, concisión, pertinencia de las respuestas, control explícito, sobrecarga cognitiva y control de errores.

Este tipo de evaluación es mixta y no permite la interacción dinámica necesaria en los sistemas de diseño cooperativos.

Este modelo de evaluación mixto al tener en cuenta la **tarea** a realizar, consigue trazar el camino que ha de seguir el usuario de manera abstracta, para después concretarlo en las acciones, necesariamente multimodales, que lo han de concretar.

4 Conceptos prácticos a medir

Definida la usabilidad en términos de objetivos y criterios operativos, [11], podemos usarla en

Funcionalidad	¿Puede el usuario realizar las tareas requeridas?
Comprensibilidad	¿Comprende el usuario el sistema?
Tiempos	¿Se realizan las tareas del usuario dentro de tiempos razonables?
Entorno	¿Encajan las tareas dentro de otras dentro del mismo entorno?
Seguridad	¿Perjudica el sistema al usuario, ya sea psicológica o físicamente?
Errores	¿Comete el usuario demasiados errores?
Comparaciones	¿Es comparable el sistema con otras formas de hacer las mismas tareas?
Estándares	¿Es el sistema análogo a otros que el usuario conozca?

forma práctica (*empírica*) pudiendo llegarse a especificar posibles medidas numéricas. Al ser muy amplios y complejos los conceptos de los que se habla en la usabilidad, es más conveniente el uso de medidas cualitativas, como las de la Tabla 4, que expresan la usabilidad del sistema en términos cualitativos y que incluyen más factores. De esta forma al menos se puede clasificar y describir el estado de los sistemas, que puedan hacer productos más usables, aunque caen en la idea reduccionista de asumir que el todo es la suma de las partes. La descomposición del problema en partes ayuda a comprenderlo pero no nos garantiza que el conjunto siga las propiedades parciales. Un producto es usable si satisface al usuario en las tareas para las que fue diseñado. Sin embargo, la medición de la satisficibilidad del usuario es una tarea difícil ya que es una función que depende no sólo de los usuarios, también de las tareas y también del entorno.

Hemos llegado pues a una definición de trabajo, suficientemente práctica para poder hacer pruebas de usabilidad. Para hacer estas pruebas prácticas será necesario incluir al usuario en las pruebas. Se deben hacer observaciones de gente haciendo cosas con el producto y tendremos que juzgar cómo de bien se están realizando las tareas. Podemos expresar esta *evaluación práctica de la usabilidad* mediante tres principios:

- las pruebas de usabilidad exigen la participación del propio usuario
- las pruebas de usabilidad se hacen mediante la observación de la realización por parte del usuario de las tareas utilizando el sistema a medir
- las medidas de la usabilidad son imprecisas y no hay prescripción que nos diga cómo es de usable un sistema. La interpretación de las observaciones siempre requieren juicios a posteriori y variarán dependiendo de las circunstancias.

Definiremos aquí métricas de la usabilidad susceptibles de ser evaluadas de una manera concreta sobre una interfaz real. Los valores medidos provienen de propiedades de la interfaz que afectan en unos u otros aspectos a la usabilidad del sistema.

Dentro de los métodos experimentales, uno de los más importantes es el de la inspección de la usabilidad, debido básicamente a [10, 9] y que está cada día más en auge dentro de los métodos realistas de evaluación de las interfaces.

La inspección de la usabilidad es el término genérico dado a un rango de métodos ingenieriles para la usabilidad que han tenido un crecimiento explosivo desde la presentación en [10] de la “evaluación heurística”, que resultó tener muchos parecidos con otros métodos de evaluación, y que finalmente en [16] han sido conjuntamente denominados métodos de “inspección de la usabilidad”.

Los métodos de inspección de la usabilidad son el nombre genérico para un conjunto de métodos basados en hacer que un evaluador inspeccione o examine los aspectos relacionados con la usabilidad de una interfaz persona ordenador. Los inspectores de la usabilidad pueden ser especialistas en usabilidad pero también lo son asesores del desarrollo del software con experiencia en determinados estilos de interfaces persona ordenador, asimismo pueden serlo usuarios finales con conocimientos sobre las tareas, etc. Así, aunque los objetivos de los diferentes métodos varían ligeramente, todos se basan en el juicio del inspector.

5 Métrica de la usabilidad

Podemos distinguir dos tipos de complejidad o esfuerzo necesario para la usabilidad, las complejidades

- estática (E)
- dinámica (D)

La primera se refiere a la dificultad propia del reconocimiento de los elementos gráficos utilizados, su reparto, estructura y aparición en una pantalla o, en general, en un punto en el diálogo hombre máquina. La segunda a la fluidez en el uso, tanto en frecuencia como en tiempo, que el usuario hace de los elementos gráficos, y por ende, de los propios diálogos o páginas. Ver [4]. Comparativamente, y haciendo una *analogía* con la evaluación de la complejidad algorítmica nos podemos referir aquí a la complejidad espacial, que sería asociable a la complejidad estática, y a la complejidad temporal, que sería la dinámica. La complejidad dinámica no tiene por qué ser sencillamente una derivación, quizás a nivel

$$D = f\left(\frac{dE}{dt}\right) \quad (1)$$

con f aproximadamente lineal. Sino que, dentro de D aparecen reflejados caminos seguidos para realizar tareas, y por tanto, longitudes de secuencias en el modelo PiE, pero a la vez independientes de la comprensión de cada uno de los pasos, y únicamente como objetivo alcanzado en un tiempo dado.

La evaluación de D no exige la evaluación previa de E pero sí puede utilizar su valor para, extraer conclusiones como la frecuencia de acceso y complejidad relativa de cada componente, no fácilmente evaluables *a priori*.

Si se evalúa previamente E , se pueden hacer estimaciones de D aún sin ejecutar el programa. Estas estimaciones no serán decisivas sino que dependerán de las selecciones que determinado usuario, y aquí tendríamos que concretar el perfil del usuario en el dominio de la aplicación, puede hacer para, según su experiencia y facilidad para reconocer las metáforas utilizadas, pueda usar para llegar (con más o menos facilidad) al objetivo del cumplimiento de la tarea prevista.

Es pues, útil, conocer previamente E para, mediante alguna técnica, como las descritas mediante los métodos experimentales y/o teóricos, poder posteriormente conocer D .

Nuestra propuesta es la de incluir en el código de alto nivel responsable del funcionamiento de los elementos interactivos sensores algorítmicos para su seguimiento y trazado. Ver, para ello [4].

Sin embargo, para la evaluación de E , son necesarias dos etapas, que aunque son complejas, no requieren de más complejo aparataje del seguimiento en tiempo real, sino que se puede realizar “en dique seco”, antes de la ejecución del programa.

Estas etapas son:

1. Análisis léxico y semántico del texto fuente de alto nivel
2. Reconocimiento de situaciones particulares en diálogos
3. Aplicación de heurísticas

La primera etapa depende, en su dificultad, como es natural, del grado de abstracción y descriptividad del lenguaje utilizado para el desarrollo, pero también depende de la coherencia y de la simplicidad interactiva de los elementos gráficos interactivos (*widgets*). En este punto es necesario hacer tantas más aproximaciones, y por lo tanto, nos alejamos más de resultados precisos, en cuanto el lenguaje sea de más bajo nivel respecto a la descripción global de los procesos interactivos. Un elemento de interacción puede contener más o menos semántica, modelando así más o menos detalles del objeto del que es metáfora. Mientras más posibilidades de respuesta ofrezca el elemento, mayor será su complejidad estática. En este sentido, la comparación entre un sencillo texto estático informativo, tal y como el pie de una imagen, es mucho menor que la complejidad de una de las flechas utilizadas típicamente para el movimiento de los contenidos de una ventana (*scroll*). Siguiendo el modelo para la evaluación de la complejidad de grandes programas, a nivel de complejidad de código, desarrollado por [7] (y posteriormente muy ampliado), es fácil encontrar, considerando al humano como un *procesador de diálogos*, en comparación con la máquina

como una procesadora de código de instrucciones, que a nivel cognitivo, la complejidad o dificultad de comprensión de un diálogo estático es función directa del número de elementos interactivos existentes en el mismo. Esta relación directa, no es, sin embargo la única influencia en tal complejidad, sino que existen muchos otros términos y factores que la modifican. Particularmente podemos destacar a nivel práctico los siguientes:

1. la categoría del elemento interactivo
2. la asociación, en primer nivel de elementos

La agrupación de los elementos gráficos interactivos presentada en [4] simplifica en 4 sencillos grupos todos los elementos gráficos interactivos usuales:

\mathcal{P}	pulsables	button
\mathcal{E}	editables	entry, text
\mathcal{S}	seleccionables	listbox, menu, menubutton, checkboxbutton, radiobutton, scrollbar
\mathcal{N}	pasivos	label, image

Con esto se zanja, de una vez por todas, la complejidad de cada elemento, y basta ahora, asignar a cada grupo un valor en forma de cantidad medida y tomada de la experiencia, de la complejidad. Este valor de la complejidad de cada categoría puede ser fácilmente interpolado de los valores de las complejidades reales medidas en experimentos en los que intervengan de forma simple, de manera que no se produzca acoplamiento entre varias complejidades interactivas.

El segundo aspecto que tenemos que tener en cuenta en la medida de la complejidad estática E de un elemento o grupo de elementos interactivos es el de su “similitud” interactiva. El uso de los elementos interactivos no se produce de manera aislada, sino que, con su agrupación se construyen objetos con un significado, no especialmente nuevo, sino cuantitativamente distinto del de cada elemento individual. La aparición, por ejemplo de una matriz de botones pulsables no tiene por qué producir una complejidad estática diferente que la de una lista móvil, un menú o una lista de botones de radio

(*radiobuttons*). En todos estos casos, se trata de elegir uno de entre varios textos (o quizás imágenes) y, por lo tanto, la dificultad real para el usuario estriba en el reconocimiento de **uno** sólo de los elementos y, a partir de él, no ya del reconocimiento individual de los demás, sino de las diferencias entre ellos. Por tanto, en este caso, la complejidad del conjunto no es necesariamente la suma de las complejidades, sino que van acompañadas de una factor, o peso, que probablemente, disminuirán la complejidad neta aportada por cada uno.

Aparte de la similitud o sinergia entre los componentes de grupos sencillos semejantes, también aparecen claras sinergias, debidas a simetrías en los diálogos, a parecidos estructurales de los mismos con objetos a los que modelan (parecido, por ejemplo, a una instancia muy familiar en su dominio para el usuario), etc. que no son fácilmente previsibles a nivel automático. La complejidad estática E incluye pues un término añadido de forma global y que corresponde a esta impresión de conjunto, independiente del número de elementos:

$$E = \sum_i c_i \sum_j e_j + e \quad (2)$$

para los grupos o elementos i dentro de cada uno de los cuales puede haber series de elementos (o sólo uno) de complejidades e_j y, finalmente una complejidad global, e .

El reconocimiento lexicográfico de los grupos de elementos depende en gran medida de la expresividad del lenguaje de alto nivel usado. Particularmente algunos lenguajes contienen directivas para la agrupación especiales y, por lo tanto, fácilmente reconocibles. Los coeficientes c_i de similitud, pueden aproximarse en primera instancia a $1/n$ con n el total de elementos del grupo.

El valor de e no es directamente previsible a nivel algorítmico, sin embargo, su influencia, y su valor, pueden ser muy próximos a cero en la mayoría de los casos, en los que el programa utilizará diálogos totalmente novedosos para el usuario.

6 Implementación

El análisis lexicográfico se incorpora al entorno de desarrollo de manera que el programador o el diseñador del prototipo, con un lenguaje de alto nivel, puede de manera continua e ininterrumpida conocer el grado de complejidad que se está introduciendo en cada diálogo diseñado.

En el peor de los casos, en el que esta evaluación no puede desprestigiar el valor de e en especial y los de c_j en agrupaciones complejas *ad hoc*, novedosas para este programa y para las que no tengamos heurísticas, al menos, tendremos un valor vigía de la complejidad, con una crecimiento y decrecimiento que acompañarán al crecimiento real de E .

Por motivos de la dificultad, inconveniencia respecto al contenido y espacio disponible, no desarrollamos aquí más las consideraciones léxicas que hace el analizador de la complejidad estática. Sin embargo, creemos necesario indicar que su portabilidad, gracias a la similitud de los entornos gráficos, fuertemente estandarizados desde la aparición del sistema Apple Macintosh, debería ser muy fácil entre lenguajes de alto nivel como se ha visto con Tcl/Tk, HyperTalk, VisualBASIC, Delphi, REALBasic y derivados.

La evaluación dinámica de la complejidad sigue mecanismos muy diferentes, relacionados fundamentalmente con el trazado y seguimiento o bien *on-line* o bien en forma de grabación de pasos seguidos para su posterior análisis. Este trazado facilita enormemente el acierto del diseño ya que muestra, de una manera indirecta y difícil de prever ni siquiera mediante una evaluación empírica, que elementos, consciente o inconscientemente son más utilizados, menos o nada en absoluto, de manera que el diseñador estudie su asequibilidad y presencia, por ejemplo.

7 Conclusiones

Se muestra en este trabajo como los métodos de evaluación de la usabilidad son condición *sine qua non* la Interfaz Persona Ordenador, no podrá evolucionar a un saludable estado de ciencia deductiva y práctica. Que estos

métodos aún arrastran técnicas de la ingeniería del software que no le son aplicables, fundamentalmente por involucrar de una manera tan especial al usuario, no modelable en el proceso de ingeniería clásico. Y que los métodos empíricos propuestos en especial para la evaluación de la usabilidad, aún estando simplificados, son muy costosos y no se llegan a emplear. Se presentan en este trabajo novedosas técnicas de evaluación de la usabilidad basadas en técnicas algorítmicas y de análisis lexicográfico previo a la ejecución, y, por tanto muy baratas. Se ha visto cómo la aplicación de heurísticas no es del todo fiable, pero aún a pesar de ello, los métodos, que no son obstrusivos, mantendrán la atención del desarrollador sobre posibles crecimientos o decrecimientos de la complejidad estática de los diálogos. Se define así también una complejidad dinámica, como complemento necesario de la anterior capaz de aportar los elementos no evaluables *a priori* en la primera.

El esfuerzo se centra pues después de obtener estos valores, en la obtención de tablas de medidas para cada categoría de elemento interactivo, dentro de un dominio para cada tipo de usuario.

Referencias

- [1] Sandrine Balbo and Joëlle Coutaz. Automatic evaluation in Human Computer Interaction, 13–16 April 1993.
- [2] Stuart K. Card, Thomas P. Moran, and Allen Newell. *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1983.
- [3] Joëlle Coutaz, Daniel Salber, and Sandrine Balbo. Towards automatic evaluation of multimodal user interfaces. Technical report, Amodeus Project Document: SM/WP32, December 1993.
- [4] Juan Falgueras Cano. *Modelado y evaluación automática cooperativa de la usabilidad de interfaces de usuario*. PhD thesis, Depto. Lenguajes y Ciencias de la Computación. Universidad de Málaga, 2000.
- [5] Bipin Indurkha. *Metaphor and cognition: and interactionist approach*. Studies in Cognitive Systems. Kluwer, 1992.

- [6] David E. Kieras. *Towards a Practical GOMS Model Methodology for User Interface Design*, pages 135–157. Number 7 in *I. Models and Theories of Human-Computer Interaction*. North-Holland, Elsevier Science Publishing Company, 52 Vanderbilt Avenue, New York, NY 10017, 1988. ISBN 0444705368.
- [7] T. McCabe. A complexity measure. *IEEE Transactions on Software Engineering*, December 1976.
- [8] Jakob Nielsen. *Usability Engineering*. Academic Press, 1993.
- [9] Jakob Nielsen and Robert L. Mack, editors. *Usability Inspection Methods*. Number 14. John Wiley & Sons, New York, 1994.
- [10] Jakob Nielsen and Rolf Molich. Heuristic evaluation of user interfaces. In *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems*, Methodology, pages 249–256. (c) Copyright 1990 Association for Computing Machinery, 1990.
- [11] David Redmon-Pyle and Alan Moore. *Graphical User Interface Design and Evaluation*. Prentice-Hall, 1995.
- [12] D. Scapin. Guide ergonomique de conception des interfaces homme-machine. Technical report, INRIA, 1986.
- [13] D. Schuler and A. Namioka. *Participatory Design: Principles and Practices*, chapter 3. , Bødker, S. and Grønbaek, K. and Kyng, M. *Cooperative Design: Techniques and Experiences from the Scandinavian Scene*. Lawrence Erlbaum, 1992.
- [14] Bernard Sufrin and Jifeng He. Specification, analysis and refinement of interactive processes. In *Formal Methods in Human-Computer Interaction*, chapter 6. Cambridge University Press, 1990.
- [15] Eileen Cornell Way. *Knowledge representation and metaphor*. Studies in Cognitive Systems. Kluwer, 1991.
- [16] Cathleen Wharton, Janice Bradford, Robin Jeffries, and Marita Franzke. Applying cognitive walkthroughs to more complex user interfaces: Experiences, issues, and recommendations. In *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems*, Usability Walkthroughs, pages 381–388. (c) Copyright 1992 Association for Computing Machinery, 1992.