

# Diseñando Software Usable

Ana M. Moreno<sup>1</sup>, Maribel Sánchez-Segura<sup>2</sup>

<sup>1</sup> Universidad Politécnica de Madrid  
Facultad de Informática  
ammoreno@fi.upm.es

<sup>2</sup> Departamento de Informática  
Escuela Politécnica Superior  
Universidad Carlos III de Madrid  
misanche@inf.uc3m.es

## 1 Motivación y Objetivos del Tutorial

Distintas clasificaciones de atributos de calidad consideran la usabilidad como un criterio de calidad más [IEEE, 98] [ISO, 00]. La usabilidad determina cuánto de fácil un sistema software es de aprender y usar, cuánto de productivos son los usuarios que trabajan con ese sistema y cuánta ayuda necesitarán [ISO, 99].

La usabilidad es una característica del software bastante compleja y que abarca un amplio espectro de aspectos. Por lo tanto, para que un sistema software sea usable es necesario ocuparse de este atributo de calidad durante prácticamente todo el proceso de construcción de dicho sistema [Ferré, 01], desde la educación de requisitos hasta las pruebas.

Una revisión en la literatura sobre HCI (Human Computer Interaction, en inglés) e Ingeniería de Usabilidad dirigido a encontrar técnicas que mejoren la usabilidad de un sistema software muestra cómo la gran mayoría de dichas técnicas se aplican a actividades relacionadas con la educación y modelización de los requisitos (técnicas de prototipado, por ejemplo), con el diseño de bajo nivel (guías de usabilidad que sugieren colores, tipos de letras, ordenamiento en la pantalla, etc.), y con la evaluación del sistema software resultante (usability testing, heuristic evaluation, etc.), siendo el diseño arquitectónico escasamente tratado.

Sin embargo, la relación entre usabilidad y arquitectura software ha sido justificada por distintos autores, tanto del campo de la HCI como de la Ingeniería de Software (IS). Por ejemplo, Ben Shneiderman, del campo de la HCI reconoce las implicaciones arquitectónicas de la usabilidad en su libro "Designing the User Interface" [Shneiderman, 98]. Por otra parte, en el campo de la IS, Bass, John y Kates del SEI [Bass, 03] han descrito un conjunto de 26 escenarios que representan características de usabilidad (por ejemplo, la capacidad de proporcionar undo o help) y han mostrado cómo contemplar dichos escenarios tiene implicaciones en la arquitectura software del sistema correspondiente. Finalmente, nosotras también hemos mostrado las implicaciones de ciertas características de usabilidad en una arquitectura software [Juristo, 03a] [Juristo, 03b].

Por este motivo, en la comunidad de desarrollo de software se está trabajando recientemente en la idea de considerar explícitamente en el diseño software determinadas características de usabilidad [Bass, 2004] [Bosch, 2003]. En este tutorial se desarrollará esta relación mediante el concepto de patrón de usabilidad sensible a la arquitectura.

Siguiendo la filosofía de los patrones de diseño [Gamma et al, 95], la idea es que los patrones de usabilidad sensibles a la arquitectura capturen la experiencia del diseño de una manera que pueda ser eficientemente reutilizada por los diseñadores, sin tener la necesidad de resolver cada problema desde cero. Al tratarse de patrones que son sensibles a la arquitectura, el objetivo es que los desarrolladores los utilicen en el momento arquitectónico, de modo que en momentos tempranos del proceso de desarrollo se puedan considerar y evaluar ciertos aspectos de usabilidad, así como plantear la relación con otros atributos de calidad.

Estos patrones proporcionarán al desarrollador una ayuda para incorporar en sus diseños determinadas características de usabilidad. Esta ayuda incluirá la solución de usabilidad que la comunidad de HCI ha definido para incorporar cada característica de usabilidad en un sistema software. Pero, además, para que dicho patrón sea realmente significativo para los desarrolladores incluirá también recomendaciones y guías para incorporar dicha característica de usabilidad en un diseño software.

En este tutorial se discutirá qué características de usabilidad son sensibles a la arquitectura (denominados, mecanismos de usabilidad sensibles a la arquitectura), así como cuál es el formato y utilización del patrón de usabilidad que ha de proporcionar al desarrollador una ayuda (tanto desde el punto de vista de usabilidad como de diseño software) para incorporar dichos mecanismos en el sistema software a construir.

## 2 Estructura del Tutorial

Este tutorial se divide en tres grandes partes, la primera se centrará en describir los conceptos fundamentales de usabilidad que un desarrollador necesita conocer, la segunda en presentar los elementos de usabilidad sensibles a la arquitectura, y la tercera en exponer y ejercitar los patrones de usabilidad que permitirán incorporar dichos elementos de usabilidad en un sistema software. Concretamente, los puntos que se tratarán son:

- Parte I. Conceptos Básicos de Usabilidad
  - Definición de usabilidad
  - Atributos de usabilidad
  - Relación entre usabilidad y arquitectura software
- Parte II. Mecanismos de Usabilidad Sensibles a la Arquitectura
  - Definición de mecanismos de usabilidad
  - Relación entre atributos de usabilidad y mecanismos de usabilidad
  - Descripción de mecanismos de usabilidad
- Parte III. Patrones de Usabilidad Sensibles a la Arquitectura
  - Patrones de usabilidad de HCI

- Formato del patrón de usabilidad sensible a la arquitectura
- Ejemplos de patrones de usabilidad sensibles a la arquitectura
- Uso de patrones de usabilidad sensibles a la arquitectura en el diseño y evaluación de sistemas software

### 3 Referencias

- [Bass, 03] L. Bass, B. John. Linking usability to software architecture patterns through general scenarios Source Journal of Systems and Software, Volume 66 , Issue 3, June 2003.
- [Bass, 04] L. Bass, N. Juristo, B. John, M. Sanchez-Segura. “Avoiding We Can’t Change It”. ICSE Tutorial, Mayo 2004.
- [Bosch, 03] J. Bosch and N. Juristo. “Designing Software Architectures for Usability”. ICSE Tutorial. Mayo 2003.
- [Ferré, 01] X. Ferré, N. Juristo, H. Windl, L. Constantine. Usability Basics for Software Developers. IEEE Software, Vol. 18 (1), pp. 22-30. Enero 2001.
- [Ferré, 03] X. Ferré, N. Juristo, A.M. Moreno, M. Sánchez-Segura. A Software Architectural View of Usability Patterns. INTERACT 2003. 2nd Workshop on Software and Usability Cross Pollination: The Role of Usability Patterns. Zürich, Suiza. Septiembre 2003.
- [Gamma, 95] E Gamma, R Helm, R Johnson, J Glissades. Design Patterns. Elements of Reusable Object-Oriented Software. Addison Wesley, 1998.
- [IEEE, 98]. IEEE Std 1061: Standard for a Software Quality Metrics Methodology. IEEE, 1998.
- [ISO, 99] ISO. ISO 13407.Human-Centred Design Processes for Interactive Systems. ISO, 1999.1
- [ISO, 00] ISO. ISO 9126-1 Software Engineering – product quality – part 1: Quality Model, 2000.
- [Juristo, 03a] N. Juristo, M. Lopez, A.M. Moreno, M. Sanchez-Segura. Workshop on Bridging the Gaps between Software Engineering and Human-Computer Interaction at ICSE '03. Portland USA, May 2003.
- [Juristo, 03b] N. Juristo, A.M. Moreno, M. Sanchez-Segura. Deliverable 3.4 Techniques, patterns and styles for architecture-level usability improvement. Abril 2003. [Http://www.ls.fi.upm.es/STATUS](http://www.ls.fi.upm.es/STATUS).
- [Shneiderman, 98] B. Shneiderman. Designing the User Interface: Strategies for Effective Human-Computer Interaction. Addison-Wesley, 1998.