

AMBIENTES DE DESARROLLO DE SOFTWARE BASADOS EN PATRONES DE USABILIDAD

Merlino, H., Vranić, A., Rodríguez, D., Pytel, P., García-Martínez, R.

Grupo Investigación en Sistemas de Información

Departamento Desarrollo Productivo y Tecnológico. Universidad Nacional de Lanús.

29 de Septiembre 3901 (1826) Remedios de Escalada, Lanús. Argentina. Tel +54 11 6322-9200 Ext. 194

rgarcia@unla.edu.ar

CONTEXTO

Este proyecto se desarrolla en el marco del Grupo de Investigación en Sistemas de Información (GSI) del Departamento de Desarrollo Productivo y Tecnológico (DDPyT) de la Universidad Nacional de Lanús con código "Proyecto UNLa 33B066: Ambientes de Desarrollo de Software Basados en Patrones de Usabilidad" con financiamiento del DDPyT y de la Secretaria de Ciencia y Técnica de la Universidad.

RESUMEN

Los patrones de diseño (design patterns) son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

Los patrones de diseño pretenden: [a] proporcionar catálogos de elementos reusables en el diseño de sistemas software, [b] evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente, [c] formalizar un vocabulario común entre diseñadores de software, [d] estandarizar el modo en que se realiza el diseño de software y [e] facilitar el aprendizaje de las nuevas generaciones de diseñadores de software condensando conocimiento ya existente.

El proyecto tiene como objetivo general definir un conjunto de patrones de software que constituirán un ambiente software para dar soporte a los diversos mecanismos de usabilidad existentes en las interfaces humano computador.

Entre los objetivos específicos se establecen: [a] caracterizar los mecanismos de usabilidad, [b] identificar los problemas que habitualmente suceden cuando se desarrolla software no

centrado en el uso de mecanismos de usabilidad, [c] identificar características similares de los mecanismos de usabilidad, [d] determinar la fiabilidad de los mecanismos de usabilidad comunes a todo desarrollo, [e] establecer un proceso que permita identificar características en los sistemas que haga necesario la utilización los mecanismos de usabilidad definidos. Se espera como aporte original de ese proyecto la generación de un conjunto de patrones de usabilidad y mecanismos para identificar situaciones propicias para su utilización.

FUNDAMENTACION

Los patrones de diseño (design patterns) son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

Los patrones de diseño pretenden:

- Proporcionar catálogos de elementos reusables en el diseño de sistemas software.
- Evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente.
- Formalizar un vocabulario común entre diseñadores.
- Estandarizar el modo en que se realiza el diseño.

- Facilitar el aprendizaje de las nuevas generaciones de diseñadores condensando conocimiento ya existente.

Uno de los aspectos que abarca la Ingeniería del Software (IS) es la de desarrollar interfaces de usuarios para que los sistemas puedan interactuar con los seres humanos; esta área de IS se denomina interfaz humano computador (HCI, de sus siglas en inglés).

La usabilidad de un producto ha sido definida en la norma ISO 9241 como, la característica que se le agrega a un producto determinado para que este pueda ser utilizado en forma efectiva, eficientemente y satisfactoria por parte de quien utilice este producto.

En este contexto, se puede definir el HCI (Human Computer Interaction) como una ciencia aplicada, la cual provee un conjunto de teorías y métodos para ser utilizados en el desarrollo de software [Carroll, 2003]

Gran parte del trabajo que se ha llevado a cabo en HCI se basa en el desarrollo de modelos predictivos los cuales se encargan de estimar el tiempo que le consumirá al usuario entender la interfaz sin tener un conocimiento previo de esta.

También se han desarrollados modelos descriptivos, estos dan una descripción detallada del modelo de solución del problema [Carroll, 2003]

Existen métodos para predecir el impacto que tendrá sobre el diseño la inclusión de factores de usabilidad, este es el denominado GOMS (Goals, Operators, Methods, and Selection rules)

Se han detallado modelos de interfaz para distintos dispositivos desde PC a teléfonos celulares, descripción de tipografías y gráficos [Cogswell, 2004]

Se ha identificado la complejidad de obtener un conjunto completo de requerimientos de software, y se ha extrapolado esto a los requerimientos de usabilidad [Ebert, et al., 2005]

Se ha tratado el tema de las implicancias particulares que tiene la edición de requerimientos para un diseño que incluya la usabilidad como parte importante de este [Juristo, et al., 2006]

Se ha definido a los patrones de usabilidad como mecanismos que pueden ser aplicados en el diseño de sistemas para proporcionar una particularidad específica de la usabilidad [Ferre, et al., 2004].

También se ha trabajado exhaustivamente en el diseño de HCI desde el punto de vista gráfico [Van Welie, 2008], existen sitios en Internet donde se detallan como deberían ser las interfaces [Yahoo!, 2008] [GNOME, 2008]

Existe un trabajo precursor [Juristo et al., 2003] sobre el tema de utilización de mecanismos de usabilidad basados en patrones arquitectónicos de software. En el mismo se detallan un conjunto de patrones que podrían ser utilizados para dar soporte a los mecanismos de usabilidad, se propone un proceso inductivo para definir dichos patrones, y se detallan los procesos de validación de los mismos. En el apartado de futuras líneas de investigación del trabajo citado se menciona la necesidad de transferir este modelo teórico a un modelo práctico para su verificación.

En este contexto surge la idea de llevar de un modelo arquitectónico de software a un modelo de patrones de diseño y de ser aplicable de lenguaje.

OBJETIVOS E HIPOTESIS DE INVESTIGACION

Este proyecto se basa en la hipótesis que muchos de los aspectos de usabilidad de patrones de diseño de software deben ser considerados como requerimientos funcionales de un sistema software. Estos deben ser lo suficiente claros para que puedan ser incluidos en los sistemas [Juristo, et al., 2007]. Por esto, surge la necesidad de abordar la identificación de estos y la creación en la medida de lo posible de patrones que los contengan.

El objetivo general de este proyecto es definir un conjunto de patrones de software que constituirán un ambiente software para dar soporte a los diversos mecanismos de usabilidad existentes en las interfaces humano computador.

Los objetivos específicos son: [a] caracterizar los mecanismos de usabilidad, [b] identificar los problemas que habitualmente suceden cuando se desarrolla software no centrado en el uso de

mecanismos de usabilidad, [c] identificar características similares de los mecanismos de usabilidad, [d] determinar la fiabilidad de los mecanismos de usabilidad comunes a todo desarrollo, y [e] establecer un proceso que permita identificar características en los sistemas que haga necesario la utilización los mecanismos de usabilidad definidos.

FUNDAMENTACION Y PERTINENCIA

Los patrones de diseño de software enfocan soluciones a problemas similares con un importante nivel de calidad. Los patrones de diseño buscan principalmente: aumentar la cohesión, reducir el acoplamiento y reducir el uso de la herencia entre objetos de software, por el alto coste que implica.

Utilizar patrones puede mejorar el diseño de una aplicación siempre que se sepa lo que se está haciendo y no se intente “meter” patrones a toda costa. Utilizar patrones aporta nomenclatura, es decir, varias personas que conozcan la terminología pueden hablar de un diseño complejo sin confusiones y sin necesidad de utilizar gráficos o esquemas, al menos cuando se tratan cuestiones generales o durante la fase inicial.

Muchos de los aspectos de la usabilidad de patrones de diseño deben ser considerados como requerimientos funcionales de un sistema software, los que deben ser lo suficiente claros para que puedan ser incluidos en los sistemas. [Juristo, et al., 2007]. Por esto se abordara la identificación de estos y la creación en la medida de lo posible de patrones que los contengan.

METODOLOGIA DE TRABAJO

En este proyecto se utilizara la metodología de desarrollo basada en prototipos. El modelo de desarrollo basado en prototipos tiene como objetivo de ayudar a comprender los requisitos del usuario. El prototipo debe incorporar un subconjunto de funciones requerida al software, de manera que se puedan apreciar mejor las características y posibles problemas. El problema del prototipo es la elección de las funciones que se desean incorporar, y cuáles son las que hay que dejar fuera, pues se corre el riesgo de incorporar características secundarias,

y dejar de lado alguna característica importante. Una vez creado el prototipo, se trabaja con él durante un período de tiempo, y a partir de la experiencia aportar nuevas ideas, detectar fallos, entre otros. Cuando se acaba la fase de análisis del prototipo, se refinan los requisitos del software, y a continuación se procede al comienzo del desarrollo a escala real. La metodología de prototipado de software se compone de las siguientes fases:

- 1.- Análisis preliminar y especificación de requisitos
- 2.- Diseño, desarrollo e implementación del prototipo
- 3.- Prueba del prototipo
- 4.- Refinamiento iterativo del prototipo
- 5.- Refinamiento de las especificaciones de requisitos
- 6.- Diseño e implementación del sistema final

Existen tres sub-metodologías de prototipado:

- Metodología basada en Prototipado desechable. Se usa para ayudar al usuario a identificar los requisitos de un nuevo sistema. En el prototipo se implantan sólo aquellos aspectos del sistema que se entienden mal o son desconocidos. El usuario, mediante el uso del prototipo, descubrirá esos aspectos o requisitos no captados. Todos los elementos del prototipo serán posteriormente desechados.
- Metodología basada en Prototipado de Maquetas. Aporta al usuario ejemplo visual de entradas y salidas. La diferencia con el anterior es que en los prototipos desechables se utilizan datos reales, mientras que las maquetas son formatos encadenados de entrada y salida con datos simples estáticos.
- Metodología basada en Prototipado evolutivo. Es un modelo de trabajo del sistema propuesto, fácilmente modificable y ampliable, que aporta a los usuarios una representación física de las partes claves del sistema antes de la implantación. Una vez definidos todos los requisitos, el prototipo evolucionará hacia el sistema final. En los prototipos evolutivos, se implantan aquellos requisitos y necesidades que son claramente

entendidos, utilizando diseño y análisis en detalle así como datos reales.

En este proyecto se utilizará metodología basada en prototipado evolutivo.

RESULTADOS OBTENIDOS/ESPERADOS

Se espera como aporte original de ese proyecto la generación de un conjunto de patrones de usabilidad y mecanismos para identificar situaciones propicias para su utilización.

Se prevee la transferencia en los siguientes ámbitos: [a] Patrones de interfaces de usuario; esto es, aquellos que intentan definir las mejores formas de construir interfaces hombre-máquina, [b] Patrones para la construcción de sistemas empresariales, en donde se requieren especiales esfuerzos en infraestructuras software y un nivel de abstracción importante para maximizar factores como la escalabilidad o el mantenimiento del sistema, [c] Patrones para la integración de sistemas, es decir, para la intercomunicación y coordinación de sistemas heterogéneos y [d] Patrones de workflow, esto es para la definición, construcción e integración de sistemas abstractos de gestión de flujos de trabajo y procesos con sistemas empresariales en el marco del BPM .

FORMACIÓN DE RECURSOS HUMANOS

El grupo de trabajo se encuentra formado por dos investigadores formados, dos investigadores en formación y un supervisor científico. En el marco de este proyecto se están desarrollando: una tesis doctoral y una tesis de maestría.

BIBLIOGRAFÍA

- Alexander, Christopher. *A Pattern Language: Towns, Building, Construction*. Oxford University press, 1977.
- Bass, Len; John, Bonnie. Linking usability to software architecture patterns through general scenarios. *Journal of Systems and Software*. Pages: 187 – 197, 2003.
- Bias, Randolph (Editor). *Cost-Justifying Usability*, Second Edition: An Update for the Internet Age, Second Edition, 2005.
- Bosch, Jan; Juristo, Natalia. Designing software architectures for usability. *Proceedings of the 25th International Conference on Software Engineering*, 2003.
- Brighton University. The Brighton Usability Pattern Collection. <http://www.cmis.brighton.ac.uk/research/patterns/home.html>
- Cogswell, Jeff. *Design Highly Useable software*. Sybex. 2004.
- Carroll, John (Editor). *HCI Models, Theories and Frameworks*. Morgan Kaufmann, 2003.
- Constantine, Larry; Lockwood, Lucy. *Software for Use: A practical Guide to the Models and Methods of Usage-Centered design*. Addison-Wesley, 1999.
- Creswell, J. 2003. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. Sage Publications.
- Dumas, Joseph. Redish, Janice. *A Practical Guide to Usability Testing*. Intellect Ltd; Rev Sub edition, 1999.
- Ebert, Chistof.; Man, Jozef.; *Requeriments Uncertainly: Influencing Factors and Concrete Improvements*. International Conference on Software Engineering 2005, pp.553-560
- Eriksson, Hans; Penker, Magnus. *Business Modeling with UML. Business patterns at work*. John Wiley and Sons, 2000.
- Ferre, Xavier; Juristo, Natalia; Moreno, Ana; Sánchez, Isabel. *A Software Architectural View of Usability Patterns*. Facultad de Informática, Universidad Politécnica de Madrid, España, 2004.
- Ferre, Xavier; Juristo, Natalia; Moreno, Ana; Sánchez, Isabel. *Framework for integrating usability practices into the software process*. Universidad Politécnica de Madrid, España, 2005.
- Ferre, Xavier; Juristo, Natalia; Moreno, Ana. *Proyecto Status*. <http://www.ls.fi.upm.es/status>. Information Societies Technology Programme, 2004
- GNOME Usability Project. <http://live.gnome.org/UsabilityProject>
- HCI Patterns Organización. <http://www.hcipatterns.org/tiki-index.php>
- Juristo Natalia, López Marta, Moreno, Ana, Sánchez M. *Techniques and Patterns for Architecture-Level Usability Improvements*. Deliverable 3.4. STATUS project, 2003. <http://www.ls.fi.upm.es/status>.
- Juristo, Natalia; Lopez, Marta; Sanchez-Segura, Maria Isabel. *Improving software usability through architectural patterns*. Universidad Politécnica de Madrid, España, 2004.
- Kumar, R.; 1996; *Research Methodology: A Stepby- Step Guide for Beginners*. Addison Wesley.
- Creswell, J. 2003. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. Sage Publications.
- Marczyk, G., DeMatteo, D., Festinger, D.; 2005; *Essentials of Research Design and Methodology (Essentials of Behavioral Science)*. John Wiley & Sons.
- Juristo, Natalia; Moreno, Ana; Sanchez-Segura, Maria Isabel. *Guidelines for Eliciting, Usability Functionalities*. Universidad Politécnica de Madrid, España, 2006.
- Juristo, Natalia; Moreno, Ana; Sánchez-Segura, Maribel; Davis. *Gathering Usability Information Through Elicitation Patterns*. Universidad Politécnica de Madrid, España, 2007.
- Tichy, W. F. *Improving the experimental method in software engineering research*. In *Experimental Software Engineering Issues*, International Workshop, Rombach and Basili and Selby (Eds.), LNCS 706, Springer-Verlag (1992), pp. 144--148.
- Van Welie, Martin. *Pattern Library*. <http://www.welie.com/patterns/>

Yahoo! Design Pattern Library. <http://developer.yahoo.com/ypatterns/>