
ABSTRACT

Building Accessible Web applications is nowadays a must. Every day more and more users with different abilities and/or temporally or permanent disabilities are accessing the Web, and many of them have special difficulties in reaching the desired information. However, the development of this kind of Web software is complicated for several reasons. Though some of them are technological, the majority are related with the need to compose different and, many times, unrelated design concerns which may be functional as in the case of most of the specific application's requirements, or non-functional such as Accessibility itself.

Even though, today there is a huge number of tools and proposals to help developers assess Accessibility of Web applications, looking from the designer perspective, there is no such a similar situation. It seems that creating accessible Web sites is more expensive and complicated than creating Web sites and then assessing/modifying them. Although this feeling may be largely true, the benefits of modelling Accessibility at early design stages outweigh the needs of a developer to implement that Accessibility.

In this thesis, we present a novel approach to conceive, design and develop Accessible Web applications in an Aspect-Oriented manner. In order to reach our goal, we provide some modeling techniques that we specifically developed for handling the non-functional, generic and crosscutting characteristics of Accessibility as a quality factor concern. Specifically, we have enriched User Interaction Diagrams (UIDs) with *integration points*, which are used to reason and document Accessibility for activity modeling during user interface design. Then by instantiating a Softgoal Interdependency Graph (SIG) *template* with *association tables*, we work on an abstract interface model (composed by ontology widgets) to obtain a concrete and accessible interface model for the Web application being developed. We use a real application example to illustrate our ideas and point out the advantages of a clear separation of concerns throughout the development life-cycle.

RESUMEN

Desarrollar aplicaciones Web Accesibles es en la actualidad una necesidad. Cada día más y más usuarios con capacidades diferentes y/o discapacidades temporales o permanentes acceden a la Web, y muchos de ellos tienen dificultades especiales para obtener la información deseada. Sin embargo, el desarrollo de este tipo de software Web es complicado por varias razones. Si bien algunas de estas son de índole tecnológicas, la mayoría están relacionadas con la necesidad de componer *intereses* de diseño distintos y muchas veces no relacionados entre sí, los cuales a su vez pueden ser funcionales, como lo son la mayoría de los requerimientos específicos de una aplicación, o no-funcionales, como lo es la Accesibilidad.

Aún existiendo hoy en día un gran número de herramientas y propuestas para ayudar a los desarrolladores en la evaluación de la Accesibilidad de las aplicaciones Web, la situación no es la misma al observar desde la perspectiva del diseñador Web. Parece ser que diseñar sitios Web accesibles es más costoso y complejo que crear sitios Web y luego evaluarlos/modificarlos. A pesar de que este sentimiento puede ser ciertamente verdadero, los beneficios al modelar la Accesibilidad en etapas tempranas del diseño superan ampliamente las necesidades de un desarrollador al implementar esa Accesibilidad.

En esta tesis, presentamos un enfoque original para concebir, diseñar y desarrollar aplicaciones Web Accesibles con una modalidad Orientada a Aspectos. Para alcanzar nuestro objetivo, ofrecemos algunas técnicas de modelado que desarrollamos específicamente para manejar las características no-funcionales, genéricas y transversales de la Accesibilidad como un *interés* de factor de calidad. Específicamente, enriquecimos los “User Interaction Diagrams” (UIDs) con *puntos de integración*, los cuales usamos durante el diseño de la interfaz de usuario, para razonar y documentar la Accesibilidad en la actividad de modelado. Luego, instanciando la plantilla del “Softgoal Interdependency Graph” (SIG) con las *tablas de asociación*, trabajamos en el modelo de interfaz abstracta (compuesta por “ontology widgets”) para obtener un modelo de interfaz concreta y accesible de la aplicación Web en desarrollo. Para ilustrar nuestras ideas y señalar las ventajas de una clara separación de *intereses* durante el ciclo de vida de desarrollo, utilizamos un ejemplo de aplicación real.