User Interface Adaptation of Web-Based Services on the Semantic Web

Nikolaos Partarakis[1], Constantina Doulgeraki[1], Asterios Leonidis[1], Margherita Antona[1], and Constantine Stephanidis[1,2]

[1] Foundation for Research and Technology – Hellas (FORTH)
Institute of Computer Science
GR-70013 Heraklion, Crete, Greece
[2] University of Crete, Department of Computer Science, Greece
cs@ics.forth.gr

Abstract. The Web is constantly evolving into an unprecedented and continuously growing source of knowledge, information and services, potentially accessed at by anyone anytime, and anywhere. Yet, the current uptake rates of the Web have not really reached their full potential, mainly due to the design of modern Web-based interfaces, which fail to satisfy the individual interaction needs of target users with different characteristics. A common practice in contemporary Web development is to deliver a single user interface design that meets the requirements of an "average" user. However, this "average" user is in fact an imaginary user. Often, the profiles of a large portion of the population, and especially people with disability, elderly people, novice users and users on the move, differ radically. Although much work has been done in the direction of providing the means for the development of inclusive Web-based interfaces that are capable to adapt to multiple and significantly different user profiles, the current evolution towards the semantic web poses several new requirements and challenges for supporting user and context awareness. Building upon existing research in the field of semantics-based user modeling, this paper aims to offer potential new directions for supporting User Interface Adaptation on the Semantic Web. In this context, the benefits gained from supporting semantically enabled ontology based profiling are highlighted, focusing on the potential impact of such an approach to existing UI adaptation frameworks.

1 Introduction

Recently, computer-based products have become associated with a great amount of daily user activities, such as work, communication, education, entertainment, etc. Their target population has changed dramatically. Users are no longer only the traditional able-bodied, skilled and computer-literate professionals. Instead, users are potentially all citizens of the emerging Information Society, and demand customised solutions to obtain timely access to any application, irrespective of where and how it runs. At the same time, the type and context of use of interactive applications is radically changing (e.g., personal digital assistants, kiosks, cellular phones and other

C. Stephanidis (Ed.): Universal Access in HCI, Part II, HCII 2009, LNCS 5615, pp. 711–719, 2009.

network-attachable equipment). This progressively enables nomadic access to information [15].

In computing, the notion and importance of *adaptation*, as the ability to adapt a system to the user's needs, expertise and requirements was only recently recognised. In this context the computationally empowered environment can adapt itself, at various degrees, to its 'inhabitants', thereby reducing drastically the amount of effort required from the users. Methods and techniques for user interface adaptation meet significant success in modern interfaces, but most focus mainly on usability and aesthetics. The Unified User Interfaces methodology for UI adaptation [15] was conceived and validated as a vehicle to efficiently and effectively address, during the interface development process, the accessibility and usability of UIs to users with diverse characteristics, supporting also technological platform independence, metaphor independence and user-profile independence.

Web-based user interfaces (WUIs) constitute a particular type of UIs that accept input and provide output by generating web pages that are transported via the Internet and are viewed by the user through a web browser. Adaptive Web-Based User Interfaces support the delivery of *qualitative user experience for all*, regardless of the user's (dis)abilities, skills, preferences, and context of use. In the web context, factors such as visual experience and site attractiveness, quality of navigation organization (especially on large sites), placement of objects [3], colour schema, and page loading time also affect the overall user experience and satisfaction and can be employed by adaptation mechanisms to personalize web user interfaces. On the other hand, the Semantic Web provides valuable means and raises great expectations WUIs adaptation. Research has already employed the features offered by the Semantic Web for generating adaptation recommendations using mining techniques [12]. In the same context, work has been conducted towards providing dynamically generated Web content to better meet user expectations through semantic browsing of information [8]. However, the potential of developing an adaptive web-based environment in the context of the Semantic Web has not yet been fully investigated. In this paper, a potential architecture for a development framework that supports the creation of adaptive Web User Interfaces is introduced by extending the architecture of an existing development framework (EAGER [5]).

This paper is structured as follows. Section 2 discusses various approaches to User Interface Adaptation. In section 3, a potential architecture for supporting User Interface adaptation on the Semantic Web is presented, based on the experience gained through the development of adaptive applications in various contexts. Section 4 outlines the main potential benefits of employing such a methodology in a semantically enabled environment. Finally, section 5 discusses further research and development steps in this direction.

2 Current Approaches to User Interface Adaptation

2.1 User and Context Profiling

The scope of user profiling is to provide information regarding the user who accesses an interactive application. A user profile contains attributes either specified by the user prior to the initiation of interaction or acquired by the system during interaction (through interaction monitoring). On the other hand, context profiling aims at collecting context attribute values (machine and environment) that are (potentially) invariant, meaning unlikely to change during interaction, (e.g., peripheral equipment or variant), or dynamically changing during interaction (e.g., due to environment noise, or the failure of particular equipment, etc).

Static profiling. Static profiling entails the complete specification of attributes prior to the implementation of the reasoning engine of an interactive application. Where static profiling is employed, the process of altering the logic used for generating the adaptable behaviors of the system is semi-automatic and cannot be provided on the fly. More specifically it is not feasible, when such an approach is followed, to enrich the decision logic while the system is running to perform meta-adaptation. This can only be achieved in the cont**ext** of adaptations that occur based on collecting and analyzing usage data.

Extensible profiling using special purpose languages and Design Support Tools. A potential solution to the limitations of static profiling is to separate the logic under which adaptation occurs from the system performing the adaptation. This can be achieved, for example, through the creation of special purpose languages for the specification of the decision logic. An example of such a language is the Decision Making Specification language (DMSL [14]). Special purpose design support tools, such as MENTOR [1], can be used to produce the decision logic of an application orchestrating user interaction.

2.2 User Interface Adaptation Toolkits

Data stemming from user and context profiling are used by adaptation toolkits for dynamically generating the interface instance that is more appropriate for a specific user in a specific context of use. Such toolkits in their most advanced implementation consist of collections of alternative interaction elements mapped to specific user and context parameters. The automatic selection of the appropriate elements is the key for supporting a large amount of alternative interface instantiations. In the following sections some indicative examples of existing tools that support the development of adaptive User Interfaces in various contexts are presented.

The EAGER toolkit. EAGER [5] is a development toolkit that allows Web developers to build adaptive applications using facilities similar to those offered by commonly user frameworks (such as ASP.NET [2] and Java server faces [6]). It is a developer framework build over ASP.NET providing adaptation-enabled ready to use dialogs. By means of EAGER, a developer can produce Web portals that have the ability to adapt to the interaction modalities, metaphors and UI elements most appropriate to each individual user, according to profile information containing user and context specific parameters.

Advanced toolkit for UI adaptation in mobile services. The main concept of this toolkit [9] is to facilitate the implementation of adaptive-aware user interfaces for mobile services. UI widgets supported by this framework encapsulate all the necessary information and are responsible for requesting and applying the relative decisions. The Toolkit employs DMSL to allow UI developers to turn hard-coded values of lexical attributes to adapted UI parameters specified in an external preference file. As a result, the UI Implementation is entirely relieved from adaptation-related conditionality, as the latter is collected in a separate rule file.

2.3 Case Studies

In this section real life applications developed utilizing adaptation toolkits are briefly overviewed, focusing on highlighting their ability to cope with the diversity of the target user population and therefore offering *qualitative user experience for all*, regardless of the user's (dis)abilities, skills, preferences, and context of use.

The AVANTI web Browser. The AVANTI Web Browser [16] facilitates static and dynamic adaptations in order to adapt to the skills, desires and needs of each user including people with visual and motor disabilities. The Avanti's unified interface can adapt itself to suit the requirements of three user categories: able-bodied, blind and motor impaired. Adaptability and adaptivity are used extensively to tailor and enhance the interface respectively, in order to effectively and efficiently meet the target of interface individualisation for end users. Additionally, the unified browser interface implements features, which assist and enhance user interaction with the system. Such features include enhanced history control for blind and sighted users, link review and selection acceleration facilities, document review and navigation acceleration facilities, enhanced intra-document searching facilities etc.

The EDEAN portal. EDEAN is a prototype portal developed, as proof-of-concept, following the UWI methodology by means of the EAGER toolkit [5]. In order to elucidate the benefits of EAGER, an already existing portal was selected and redeveloped from scratch. In this way, it was possible to identify and compare the advantages of using EAGER, both at the developer's site, in terms of developer's performance, as well as at the end-user site, in terms of the user-experience improvement.

The ASK-IT interface for mobile transportation services. The Home Automation Application developed in the context of ASK-IT facilitates remote overview and control through the use of a portable device. These facilities provided the ability to adapt themselves according to user needs (vision and motor impairments), context of use (alternative display types and display devices) and presence of assistive technologies (alternative input devices).

2.4 Discussion

The approaches developed so far to support User Interface Adaptation have shown to be adequate for addressing a number of requirements. Especially in the context of web applications, previous work has proven that it is technologically feasible to develop web-based interfaces that are able to adapt to various user profiles and contexts of use. Limitations of current approaches include the difficulties faced when addressing the potential of change and the reduced reasoning capabilities resulting from the methods used for capturing user and context profiles. The semantic web brings new directions and challenges, and offers new paths through enhanced expressive power and advanced reasoning facilities. The next section discusses how these facilities can be used towards enriching the adaptive behavior of existing frameworks. A potential implementation architecture for supporting User Interface Adaptation on the Semantic Web will be presented, focusing on the feasibility of such a concept and on its potential advantages.

3 User Interface Adaptation on the Semantic Web

3.1 Requirements for Effective User Modeling

Requirements for creating effective user modeling systems have been documented in [7] and [4], and include:

- **Generality, including domain independence.** User modeling systems should be usable in as many domains as possible, and within these domains for as many user modeling tasks as possible.
- **Expressiveness and strong inferential capabilities.** Expressiveness is a key factor in user modeling systems; they are expected to express many different types of assumptions about the users and their context. Such systems are also expected to perform all sorts of reasoning, and to perform conflict resolution when contradictory assumptions are detected.
- **Support for quick adaptation.** Time is always an important issue when it comes to users; User modeling systems are required to be adaptable to the users' needs. Hence they need to be capable of adjusting to changes quickly.
- **Precision of the user profile.** The effectiveness of a user profile depends on the information the system delivers to the user. If a large proportion of information is irrelevant, then the system becomes more of an annoyance than a help. This problem can be seen from another point of view; if the system requires a large degree of customization, then the user will not be willing to use it anymore.
- **Extensibility.** A user modeling system's success relies on the extensibility it offers. Companies may want to integrate their own applications (or API) into the available user models.
- **Scalability.** User modeling systems are expected to support many users at the same time.
- **Import of external user-related information.** User models should support a uniform way of describing users' dimensions in order to support integration of already existing data models.
- **Management of distributed information.** The ability of a generic user modeling system to manage distributed user models is becoming more and more important. Distributed information facilitates the interoperability and integration of such systems with other user models.
- **Support for open standards.** Adherence to open standards in the design of generic user modeling systems is decisive since it fosters their interoperability.
- **Load balancing.** User modeling servers should be able to react to load increases through load distribution and possibly by resorting to less thorough (and thereby less time-consuming) user model analyses.

- **Failover strategies.** Centralized architectures need to provide fallback mechanisms in case of a breakdown or unexpected situation.
- **Fault tolerance.** In case a user inserts wrong data in his/her profile by mistake (i.e. a user denotes an opposite gender), the system must prompt the user to adjust the corresponding parameters, rather than reset his/her profile.
- **Transactional Consistency.** Parallel read/write procedures on the user model should lead to the deployment of sufficient mechanisms that preserve and restore possible inconsistencies.
- **Privacy support.** Another requirement of user modeling systems is to respect and retain the user's privacy. In order to meet these requirements, such systems must provide a way for the users to express their privacy preferences, as well as the security mechanisms to enforce them

3.2 User Interface Adaptation on the Semantic Web: Proposed Architecture

Figure 1 presents the proposed implementation architecture for supporting adaptive interfaces on the Semantic Web.

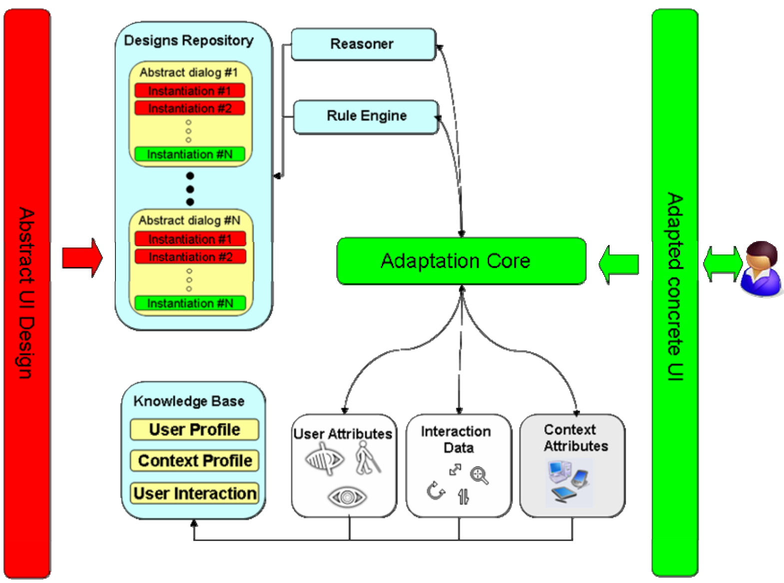

Fig. 1. User Interface Adaptation on the Semantic Web: proposed architecture

Modeling User, Context and Interaction. In the proposed architecture, the Knowledge Base contains the ontology representing the modeled classes and properties for supporting the collection of parameters appropriate for modeling:

- User Profile (Disability, Web Familiarity, Language, etc.)
- Context Profile (Input-Output devices, screen capabilities, etc.)
- User Interaction (monitoring user actions, user navigation paths, etc.)

The Knowledge Base can use web ontology languages such as OWL to store the appropriate information in the form of semantic web rules and OWL-DL [11] ontologies. This approach offers enough representational capabilities to develop a formal context model that can be shared, reused, and extended for the needs of specific domains, but can also combined with data originating from other sources, such as the Web or other applications. Moreover, currently the logic layer of the Semantic Web is evolving towards rule languages that enable reasoning about the user's needs and preferences and exploiting available ontology knowledge [10]. An example of how user profile parameters can by modeled in an ontology is presented in Figure 2. User is a superclass that includes the user groups a user may belong to according to his/her functional limitations (NonImpairedUser, HearingImpairedUser, MotorImpairedUser or VisuallyImpairedUser), each of which is further analysed where appropriate.

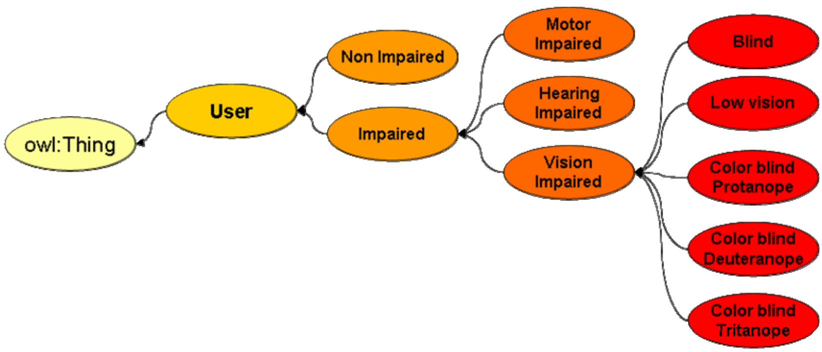

Fig. 2. An example of an ontology representing user abilities

Designs Repository. The Designs Repository contains abstract dialogues together with their concrete designs. Following the Unified User Interface Design methodology [15], this is achieved through polymorphic decomposition of tasks that leads from abstract design pattern to a concrete artifact. Design Repositories for supporting adaptation of web-based services can consists of primitive UI elements with enriched attributes (e.g., buttons, links, radios, etc.), structural page elements (e.g., page templates, headers, footers, containers, etc.), and fundamental abstract interaction dialogues in multiple alternative styles (e.g., navigation, file uploaders, paging styles, text entry) [5].

Reasoner and Rule Engine. The Reasoner module, together with the Rule engine, undertakes the job of classifying instances and performing the overall decision making that is required for selecting the appropriate interaction elements to build the concrete user interface. In this context, the Reasoner classifies instances into classes that have a strict definition, taking into account the Open World Assumption (i.e., if there is a statement for which knowledge is not currently available, it cannot be inferred if it is true or false). The Rule Engine undertakes the classification into primitive classes and specifies and executes classification rules.

Orchestration (Adaptation Core). The adaptation core undertakes the orchestration of the main modules of the proposed architecture. When a user profile is created, the Reasoner and Rule engine are invoked for classifying instances under various classes, computing inferred types and reasoning on the available context. The results are stored in the knowledge based and are used by the adaptation core for inferring specific actions regarding the activation and deactivation of alternative dialogs. The adaptation core is also responsible for re-invoking the aforementioned services when the data stemming from the user interaction monitoring process lead to the need of reevaluating existing user profile information through reevaluation of rules.

3.3 Benefits

Regarding the adaptation process itself, the adoption of a semantically enabled inference mechanism potentially allows the evaluation of more complex rules, thus making reasoning more solid and enriching the application logic. Moreover, an ontology based specification of user, context and interaction profiles makes the potential extension of the system easier. Another important benefit of a semantically enabled adaptation approach is the increased possibility of learning user preferences. These attributes traditionally can be set by the user, but in most cases cannot be inferred from user actions. In the context o the proposed architecture it is possible to dynamically generate social tags that can in turn be used for performing adaptive filtering of information based on user preferences. A similar result can be also obtained by modeling user interaction data and performing batch analysis. This can be supported in the proposed architecture through introducing another layer of modeling beyond the designs repository used for strict UI purposes (i.e., a content modeling repository).

4 Conclusions and Future Work

This paper has proposed an architecture for supporting the development of Adaptive User Interfaces on the Semantic Web, based on existing approaches which have been successfully used in the recent past for supporting adaptation of user interfaces in various contexts. Modifications to the architectural structure used in these adaptation frameworks have been proposed in order to cope with the requirements set in the context of the Semantic Web. Taking into account the enriched modelling and inference capabilities offered, this novel architecture aims at combining the benefits of the Semantic Web (such as extensibility, strong inference capabilities, etc.) with benefits of existing adaptation frameworks (such as the ability to address accessibility, user preference, various input output devices, etc).

In future work, this implementation architecture will be employed in the context of the EAGER development framework. In this context, the Knowledge Base of Eager together with the inference mechanisms will be replaced by the modules proposed in the extended architecture (Knowledge base, Rule engine, Reasoner, etc.) allowing the reuse of facilities common to both architectures, such as the Designs Repository (which has been already put into use in the context of several interactive web based applications, such as the EDEAN portal, http://www.edean.org).

References

1. Antona, M., Savidis, A., Stephanidis, C.: A Process–Oriented Interactive Design Environment for Automatic User Interface Adaptation. International Journal of Human Computer Interaction 20(2), 79–116 (2006)
2. ASP.NET Web Applications, http://msdn.microsoft.com/en-us/library/ms644563.aspx
3. Bernard, M.L.: User expectations for the location of web objects. In: Proceedings of CHI 2001 Conference: Human Factors in Computing Systems, pp. 171–172 (2001), http://psychology.wichita.edu/hci/projects/CHI%20web%20objects.pdf [2007]
4. Çetintemel, U., Franklin, M.J., Giles, C.L.: Self-adaptive user profiles for large-scale data delivery. In: Proceedings of the 16th IEEE International Conference on Data Engineering (ICDE 2000) (2000)
5. Doulgeraki, C., Partarakis, N., Mourouzis, A., Stephanidis, C.: Adaptable Web-based user interfaces: methodology and practice. eMinds 1(5) (2009) http://www.eminds.hci-rg.com/index.php?journal=eminds&page=article&op=download&path=58&path=33
6. JavaServer Faces Technology, http://java.sun.com/j2ee/1.4/docs/tutorial/doc/JSFIntro.html
7. Kobsa, A.: Generic user modeling systems. User Modeling and User-Adapted Interaction 11(1-2), 49–63 (2001)
8. Kostkova, P., Diallo, G., Jawaheer, G.: User Profiling for Semantic Browsing in Medical Digital Libraries. The Semantic Web: Research and Applications, 827–831 (2008)
9. Leuteritz, J.-P., Widlroither, H., Mourouzis, A., Panou, M., Antona, M., Leonidis, A.: Development of Open Platform Based Adaptive HCI Concepts for Elderly Users. In: Stephanidis, C. (ed.) Proceedings of 13th International Conference on Human-Computer Interaction (HCI International 2009, San Diego, California, USA, July 19–24. Springer, Berlin (2009)
10. Michou, M., Bikakis, A., Patkos, T., Antoniou, G., Plexousakis, D.: A Semantics-Based User Model for the Support of Personalized. Context-Aware Navigational Services. In: First International Workshop on Ontologies in Interactive Systems, 2008. ONTORACT 2008, pp. 41–50 (2008)
11. OWL Web Ontology Language Reference. W3C Recommendation, February 10 (2004), http://www.w3.org/TR/owl-ref/
12. Robal, T., Kalja, A.: Applying User Profile Ontology for Mining Web Site Adaptation Recommendations. In: Ioannidis, Y., Novikov, B., Rachev, B. (eds.) ADBIS 2007. LNCS, vol. 4690, pp. 126–135. Springer, Heidelberg (2007)
13. Savidis, A., Leonidis, A., Lilis, I., Moga, L., Gaeta, E., Villalar, J.L., Fioravanti, A., Fico, G.: Self-configurable User Interfaces. ASK-IT Deliverable - D3.2.1 (2004)
14. Savidis, A., Antona, M., Stephanidis, C.: A Decision-Making Specification Language for Verifiable User-Interface Adaptation Logic. International Journal of Software Engineering and Knowledge Engineering 15(6), 1063–1094 (2005)
15. Stephanidis, C.: The concept of Unified User Interfaces. In: Stephanidis, C. (ed.) User Interfaces for All - Concepts, Methods, and Tools, pp. 371–388. Lawrence Erlbaum Associates, Mahwah (2001)
16. Stephanidis, C., Paramythis, A., Sfyrakis, M., Savidis, A.: A Case Study in Unified User Interface Development: The AVANTI Web Browser. In: Stephanidis, C. (ed.) User Interfaces for All - Concepts, Methods, and Tools, pp. 525–568. Lawrence Erlbaum Associates, Mahwah (2001)