A Demonstration of Adaptive Work-Centered User Interface Technology

Amy Aragones, GE Global Research, 1 Research Circle, Niskayuna, NY, 518-387-4009, aaragones@crd.ge.com

Jeanette Bruno, GE Global Research, 1 Research Circle, Niskayuna, NY, 518-387-6009, bruno@crd.ge.com

Andrew Crapo, GE Global Research, 1 Research Circle, Niskayuna, NY, 518-387-5729, crapo@crd.ge.com

Marc Garbiras, GE Global Research, 1 Research Circle, Niskayuna, NY, 518-387-6654, garbiras@crd.ge.com

ABSTRACT

A key challenge in designing for the Semantic Web is to address open-ended decision-making scenarios. This demonstration will show the benefits of Adaptive Work-Centered User Interface Technology (ACUITy)in this regard in the context of a Professor/Student Course Management application.

Keywords

Semantic Web, Semantic Technology, Intelligent User Interface, Adaptive User Interface, Web Portal, Work-Centered Support System

1. INTRODUCTION

One of the challenges of designing applications for the Semantic Web is to create mixed interaction structures in which users and agents can collaboratively solve problems that may use a multitude of alternative, often ad-hoc decision-making strategies. The appropriateness of alternative strategies might be influenced by subtle differences in user goals, preferences and/or other context that is difficult to anticipate in the upfront design of a new web application. A second related challenge is to create an environment where new Semantic Web applications can easily be developed, particularly designing for these open-ended decision-making scenarios.

Our demonstration introduces the Adaptive Work-Centered User Interface Technology (ACUITy) decision support environment, which begins to answer these challenges. In ACUITy we use semantic models to characterize the users' work domain in terms of "work-centered"[1] activities as well as the interface and interactions between the decision support system and the user. We use these semantic models to provide adaptive interaction, both user directed and automated, in the characterization and presentation mechanisms of a web-based user interface.

The Professor/Student Course Management (PSCM) application built using ACUITy will illustrate the use of semantic technology to implement work-centered decision support and the benefits of ACUITy to both application users and developers. We will use this demonstration to highlight the broader potential for ACUITy in areas that are of interest to the Semantic Web community. The PSCM application will also accompany our release of ACUITy on open source in 2006.

Demos and Posters of the 3rd European Semantic Web Conference (ESWC 2006), Budva, Montenegro, 11th - 14th June, 2006

2. PROFESSOR/STUDENT COURSE MANAGEMENT APPLICATION

2.1 Professors' Course Management

Our demonstration focuses on two aspects of an instructor's course management work. First, we will demonstrate enhanced capabilities for monitoring and understanding the logistical aspects of a course, including the size of the course, sections, schedules, support staff and the publishing of the syllabus, including evaluation and grading formula for the course.

Second, we target the analysis and production of student grades at the end of the semester. This includes supporting the assessment of the class numerical grades (as computed by the formula published at the beginning of the semester) and establishing the numeric breakpoints for translating the students' grade averages into an A, B, C, D, or F grade.

2.2 Students' Course Management

We will also demonstrate support for a portion of the work students perform in creating a course schedule. Before a new semester begins the student must determine which courses to take and register for them. This usually involves understanding the student's course requirements and academic status relative to the University's degree requirements. It also requires understanding course offerings with respect to fulfilling the student's degree requirements and scheduling constraints.

The following sections elaborate on the capabilities we will demonstrate.

3. WHAT ACUITy MEANS TO USERS

3.1 Users Finish the Design

In an ACUITy application the users themselves finish the design of a user interface by deciding what information they require to solve a particular problem – defining the vantage they need on the problem domain – and changing the characteristics of the information display in order to interact with the data more effectively. ACUITy captures in a centralized way the experience of users in open-ended problem-solving domains as they gather information from many disjoint sources not precisely identified at design time.

The user can reconfigure the display by adding and removing displayed information. The approach can be extended to permit ad-hoc additions of information sources. Customization of information display includes, but is not limited to, the hiding, ordering, and sorting of data table columns, the selection of graph series types, e.g., line versus bar, color, and labels, and the type of enumerated selection lists, e.g., dropdown list versus checkbox versus tabs. The user can duplicate and then modify visualizations as desired. Information in disjoint tables and graphs can also be brought into relation by creating shared highlight regions, similar to data brushing in statistical graphics.

3.2 Learning Defaults and Patterns

Learning from accumulated instance data is an implicit benefit of the semantic modeling approach taken by ACUITy. As users adapt the content and visual characteristics of the information that they view in particular problem-solving settings, these changes are stored in the ontology with their context. This past history creates the opportunity for a reasoner to infer what information content is most appropriate based on new information that was unavailable during the initial design of the user interface. This special purpose reasoner then uses the instance data to learn both default content and appearance for new sessions with similar contexts. We have implemented several different algorithms to learn new default displays from instance data. We will demonstrate the effects of these algorithms and how we have implemented them using OWL.

While not yet implemented in ACUITy, recognizing beneficial patterns of usage across groups of users can lead to new classes of display objects explicitly available to developers and users, whereas learned defaults are only implicitly available. It is not inconceivable that abstraction of useful patterns might even extend across application domains.

4. WHAT ACUITy OFFERS DEVELOPERS

The ACUITy Problem-Vantage-Frame (APVF) ontology, described below, provides developers of new applications a starting point from which they can create information-rich displays by relatively simple model extensions. With respect to the user-interface, the developer is also "finishing the design." For example, a new data table can be added to a display through a few simple steps. The behavior and attributes necessary for the table to be constructed and displayed, as well as those that allow the user to customize the table display according to their particular preferences, are inherited.

Domain-specific work models also utilize upper-level ontologies in ACUITy. These ontologies define concepts of time, physical versus abstract, problems, scripts, processes, and remote data sources. Scripting capability includes support of custom Java code that can implement data access, data transformation, or side effects. This facilitates integration of ACUITy applications with existing information repositories and computational models.

5. IMPLEMENTATION APPROACH

ACUITy has three major components:

1. *The ACUITy Problem-Vantage-Frame (APVF) ontology*, which, situated within a hierarchy of upper level and domain-specific ontologies, represents concepts and properties that describe users, the problems they are trying to solve, the information they (and other users) have used to solve those types of problems and the display properties of that information. The APVF ontology is represented in OWL.
2. *The ACUITy Controller*, a Java class (with supporting classes) that provides an API to the APVF ontology. It provides special-purpose reasoning over this knowledge base to determine the set of information relevant to the problem at hand or the context of work performed. The ACUITy controller queries the ontology to understand where to find data, how to obtain it, and how to bundle it. The controller also accepts inputs from the client UI engine and updates the ontology accordingly.
3. *The User Interface (UI) Engine*, which accepts the ontological information obtained from the ACUITy Controller and creates the application's user interface. It interacts with the controller to request information from the ontology in response to the user's actions. At this point, we have implemented a web client renderer to produce a well-formed HTML document from the UI engine.

6. FUTURE DIRECTIONS

To date, ACUITy has been used to prototype several web applications of interest to General Electric, Lockheed Martin and the US Air Force. Our preliminary experience is that it is a powerful and very flexible environment for developing decision support systems. Additional user testing and developer feedback is needed to validate this initial assessment.

We plan to create - and hope to also demonstrate at ESWC - an ACUITy Ontology Editor that will simplify the process of creating and editing web applications for both developers and end users. New sources of data will become relevant to users during the course of their work, potentially via web-services, ad hoc web searching and other means of accessing information. We would like users to be able to map to these sources of information in real time.

We hope that by making this work available to the public we can contribute to the wide-spread use of semantic technology. We also believe that wider research efforts will be able to leverage data-rich models created by ACUITy that are populated with semantically tagged information. These will provide an important resource for further research in semantically-enabled UI and decision support system design, adaptation and learning.

7. ACKNOWLEDGMENTS

This work was partially funded by the Air Force Research Laboratory, Wright Patterson AFB, under contract F33615-03-2-6300. We would like to acknowledge the contributions of AFRL and Lockheed Martin STS, who provided support to and feedback on this research.

8. REFERENCES

[1] Eggleston, R.G., Young, M.J., and Whitaker, R.D. Work-Centered Support System Technology: A New Interface Client Technology for the Battlespace Infosphere. *NAECON 2000*, 499-50.