Building and Using an OWL-S Ontology of Tasks

Fatma Mili[1] and Swapna Ghanekar[1]

Oakland University, Rochester MI 48309-4478, USA,
mili@oakland.edu

1 Background, Motivation

Systems in which there is a human in the loop are hard to design, analyze, and verify. When the correct behavior of these systems is safety-critical or mission-critical, considerable effort must be invested in the design of the human-machine interface and in the analysis of the integrated human-machine behavior. Although accidents are often attributed to "operator error", these errors can often be predicted and avoided by accounting more thoroughly for the human in the loop [6]. Decisions concerning the human-machine interface are more than afterthought cosmetic decisions of layout, color, and font; they concern the essence of the functionality of the system encompassing decisions concerning which tasks to automate and which tasks to leave for the human operator and, for those tasks that are not fully automated, deciding for the support role that the system can play (see, e.g. [3]). These collaborative decisions are made taking into account the capabilities and limitations of the human agent including the speed of making complex decisions, the capacity to recall information, the accuracy with which such information is recalled, as a function of its recency—among other factors, the effect of mental overload and mental overload, and the effects of other human behavioral modifiers such as fatigue, stress, fear, and other environmental and emotional conditions [1, 2, 7–10].

The detailed qualitative and quantitative analysis of the functionality and service delivery of systems with a human in the loop are an important component of design. Performance modeling [4, 5] and cognitive modeling [1, 2] are the main approaches available to perform such analysis. The models generated are, by nature, very detailed (tasks described at the mouse click level), and thus labor intensive. Furthermore, because the models tightly integrate elements of the service at hand (e.g. select a new itinerary) with elements of the interface (point and click on an itinerary from a menu) and characteristics of the agent performing the task, they make it very difficult to reuse a model if any one of these factors changes. The lack of reuse proved to be a major impediment to the widespread use of human-centered system analysis during design.

2 Ontology of Military Tasks

In the past year, we have worked in collaboration with NIST and DC-corp in an effort to develop a repository of tasks of interest to the design of military vehicles. The motivations for this project are as follows:

- The detailed and thorough analysis of the integrated machine-human military systems is critical for the safety of the humans involved and for the success of the missions in which these vehicles are used.
- The detailed and thorough analysis of the integrated machine-human military systems allows early consideration of a variety of designs and agent allocation.
- Currently little or no analysis takes place because modeling is labor intensive and because it is very hard to reuse models under different configurations.
- Although interfaces and technology change, the core functionality of military systems is rather stable. These missions must be documented reused across platforms and across generations of vehicles.
- The documentation must allow for different views (general vs. detailed) of the mission and its component tasks and must support reasoning about the tasks based on their components.
- The documentation must be compatible with the information needed by cognitive models and performance models.

We have used the OWL-S upper ontology of services to describe tasks involved in missions. For the purposes of this paper, a task is a service offered by a system within the scope of a mission. A task has a task profile inherited from the service profile. The main additions introduced are:

Transformation Information: In addition to its input I, Output O, precondition P, and effect E, every task uses a set of resources R. Each resource has a type (e.g. cognitive, motor-left-hand), and a quantity.

Structural Information: Composite tasks inherit from composite services and thus can have any of the control constructs defined in the OWL-S ontology of services. Whereas OWL-S does not formally define the semantics of the control construct, we define a set of axioms defining the relationship between a composite task and its component and a set of rules and triggers dictating how new information gets propagated in a way that maintains the consistency of the axioms.

3 Issues and Statement of Interest

We are interested in participating in this workshop to

Share our experience with the upper ontology. We have used the OWL-S upper ontology in a domain that not related to the semantic web, yet, the OWL-S ontology proved to be a very good fit. We believe that this experience is useful to others who are contemplating developing domain models that are at the same time human readable and machine readable, that are easy to update through progressive refinement, and that call for a rich collection of composition constructs. We have used Protege as the editing tool.

Discuss the semantic definitions of the control constructs. In our application, one of the strongest requirement is the ability to reason about tasks based on their structure and the ability to progressively refine the task model and have new information propagated correctly. This reasoning requires a formal definition of the semantics of the different control constructs. For each control construct, we have defined a horizontal consistency rule, a computation procedure, and a vertical consistency rule. The horizontal consistency rule defines the conditions that must hold between the components of the same task. For example, not any pair $< T_1, T_2 >$ can form a coherent sequence. The effect of T_1 must not be in conflict with the precondition of T_2. In case the set of components meet the horizontal consistency rule, a computation procedure defines how the values of the composite construct (input, output, precondition, effect, and other parameters) are computed from the values of the components. The progressive refinement of the ontology allows simple tasks to be expanded into composite tasks. The vertical consistency rule defines the relationship that must hold between the values of the simple task and those (computed values) of the composite task it expands to.

Discuss choices made in implementing triggers For an ontology to be coherent, the horizontal and vertical consistency rules must be met at all times. All additions and changes to the ontology must be monitored to ensure that the coherence is preserved. Of particular interest, is the process of refinement through expansion of simple tasks, and through revision of previously entered values. We discuss a set of triggers that we have implemented (as Protege plugins) to react to changes, propagate values, and communicate problems to the ontology editor. We will discuss the tradeoffs between using rules that are part of the ontology and using triggers as Protege plugins (the choice we made). We will also discuss potential performance related to the triggers due to the fact that the ontology privileges navigation from composite to components over components to composite and alternative solutions to this performance issue.

References

1. John Anderson, J., Lebiere: Atomic Components of Thought. Lawrence Erlbaum. 1998.
2. Card, S., Moran, T. Newell, A: The Psychology of Human-Computer Interaction. Hillsdale, NJ: Erlbaum. 1983.
3. Mike Falla, (ed): Advances in Safety Critical Systems, dti, EPSRC
4. Mitchell, D.K., Advanced Improved Performance Research Integration Tool (IMPRINT): Vetronics Technology Test Bed Model ARL-TN-0208, September 2003.
5. John, B., Kieras, D.E.: The GOMS Family of User Interface analysis Techniques: Comparison and Contrast, ACM Transactions on Computer-Human Interaction, 3, pp.320-351, 1996.
6. Nancy Leveson.: A New Accident Model for Engineering Safer Systems. Safety Science, Vol. 42, No. 4, April 2004, pp. 237-270.
7. Nancy Leveson.: Intent Specifications: An Approach to Building Human-Centered Specifications. International Conference on Requirements Engineering, 1998.
8. Frank E. Ritter, Baxter, G. D., Jones, G., Young, R. M.: Supporting cognitive models as users. ACM Transactions on Computer-Human Interaction, 7(2), pg.141-173. 2000.

9. Frank E. Ritter, Shadbolt, N. R., Elliman, D., Young, R., Gobet, F., Baxter, G. D.: Techniques for Modeling Human and Organizational Behavior in Synthetic Environments: A supplementary Review Wright-Patterson Air Force Base, OH: Human Systems Information Analysis Center. 2003.
10. David D. Woods: Towards a theoretical base for represention design in the computer medium: Ecological perception and aiding human cognition. In J. M. Flash et al, editors. An Ecological Approach to Human Machine Systems I: A Global Perspective, Erlbaum, Hillsdale, New Jersey, 1995.