Work-Centered Services for the Semantic Web

Andrew Potter
apotter@Sentar.com
Gordon Streeter
gstreeter@Sentar.com

Sentar, Inc.
4900 University Sq., Suite 8
Huntsville, Alabama 35816 USA
Fax: 256.430.0840
Phone: 256. 430.0860

Keywords: SEMANTIC WEB, MULTI-AGENT, WORK-CENTERED

Work-Centered Services for the Semantic Web

Andrew Potter
apotter@Sentar.com
Gordon Streeter
gstreeter@Sentar.com

Sentar
Huntsville, Alabama, USA

Abstract

The Semantic Web will enable systems to communicate intelligently with one another. As this occurs, the nature and extent of human interaction with computers will alter correspondingly. Although the characteristics of interaction cannot be fully anticipated, it is clear that as systems gain intelligence and autonomy, the demands placed on them will become more exacting. To meet these expectations, the Semantic Web should incorporate work-centered services. These services will provide focused, intelligent points of contact between the Web and its users. Users will access work-centered services to formulate and conduct investigations, and to collect, organize, and analyze results. Work-centered services will direct tasks in behalf of the user and communicate with the user in a humanly coherent manner. As a result, the concept of the user interface will broaden and deepen to include specialized agents, possessing the expertise to assemble information for end-user consumption. These agents will occupy the web-continuum just as any other knowledge resources. This paper introduces the concept of work-centered services and describes related research and development being performed as part of the KnoWeb project.

Introduction

The Semantic Web will provide intelligent content for use by automated systems, enabling software services to perform distributed reasoning using Web resources (Berners-Lee, Lassila). Agents will be able to solve complex problems by gathering information from diverse sources and synthesizing a result tailored to the user's requirements. "Searching the Web," now a predominate pastime among Web users, will diminish in favor of more proactive forms of information retrieval and problem solving.

The user experience will evolve to take advantage of these new possibilities. Just as the expert systems of the eighties fostered new requirements for the user interface (Hendler, Potter), the Semantic Web brings new challenges for user interface designers. As the Web is employed to undertake tasks previously accomplished by other means (or not accomplished at all), the significant points of contact between the Web and end-users are redefined. To the extent that we may attribute intelligence to Web resources (i.e. to the extent that the Web is semantic), the reasoning processes performed on our behalf need to be visible. Further, the complexity of resources through which information may be distilled and synthesized indicates that the user will interact with information deriving from a confluence of sources. The user interface must be able to present this information coherently and credibly.

To meet these demands, the Semantic Web needs to include work-centered services. Work-centered services will act as intelligent liaisons, intelligent points of contact, between user applications and the greater Web. Users will access these services to formulate and conduct investigations, to collect, organize, and analyze results, to direct tasks in behalf of the user, and to communicate with users in a humanly coherent manner. In this paper, we introduce work-centered services and describe a system designed to tackle these issues.

Semantic User Interface

It has been suggested that the end-user counterpart to the Semantic Web might be called the "Semantic User Interface." This Semantic User Interface would "separate the functions of the UI from its visual appearance" (Saravanan). This approach constrains the concept of the user interface to the status quo, e.g. windows, standardized graphic controls, and decorations. However, this underestimates the promise of the Semantic Web. The user interface for the Semantic Web must go beyond this. It must be intelligent, distributed, and knowledge-intensive. The user interface will exist as a continuum of the rest of the Web and will include knowledge agents which know how to assemble information for end-user interaction. These agents will specialize in forms of knowledge useful in interacting with humans, such as question-and-answer, decisions trees, workflow, or media management.

We must adopt a view of the user interface that goes beyond look and feel. The direct manipulation interfaces common today have proven popular and serviceable in desktop applications, both intelligent and unintelligent, for many years. Graphic controls, such as list boxes, pull-down menus, pop-up menus, tree menus, scrollbars, and tool tips have all been used to considerable advantage. DHTML has enabled software developers to provide a fortified (if sometimes chaotic) user experience when browsing the Web. All this is necessary, perhaps, but insufficient, for delivering the Semantic Web to the desk (or palm) top. That intelligent applications place serious demands on the user interface comes as no surprise. The interface is, after all, the delivery mechanism for system functionality. Any intelligence the system may possess is all for naught if it cannot be placed at the user's disposal. For the user interface to be effective, it must draw upon the underlying capabilities of the application to enable the user to do what needs to be done.

These difficulties are rendered more challenging when the underlying application is implemented on a confederation of loosely affiliated systems, such as the Semantic Web. Here, amid the emergent realms of multi-agent systems and intelligent Web services, users will encounter amorphous, non-deterministic collections of distributed entities, conscripted on call to the user's goals and aspirations. To make this encounter interesting and beneficial to the user, a deeper understanding of the user interface is mandatory.

Work-Centered Services

Work-centered services promote productive human activity. People use the Internet for many purposes, e.g. reference, entertainment, problem solving, and time management. In order to limit the scope of our research to somewhat manageable proportions, our emphasis has been on information retrieval and problem solving. In this context, unlike non-directional Web browsing, work-centered services seek to fulfill the well-worn mantra, "the right information to the right person at the right time."[1] To accomplish this will be to solve one of the Internet's perennial problems. Work-centered technology is the convergence of two separate developments. The first of these is the development of work-centered support systems (Eggleston), which seek to reduce work complexity by integrating direct manipulation with multi-agent technology.

The second of these is the KnoWeb multi-agent architecture, a, distributed problem solving technology for the Web (Streeter, 2001). Work-centered services use KnoWeb technology to reduce the complexity of productive human activity on the Web. This technology is applicable to numerous

[1] For example, http://www.research.ibm.com/knowsoc/stories_german.html

problems. Prototypes include a multi-agent customer support application, a product recommendation tool for use in a customer resources management system, and an event-driven decision support system.

Work-centered services are not distinct from information resources *per se.* They are enabling resources that help users access content in an intelligible, useful manner. Figure 1 shows how work-centered services interpose between the user and information resources to achieve this. The designation of these services as work-centric is functional rather than architectural. Architecturally, they share the same stature and define the same interfaces as knowledge agents. The difference is in the role they play. They specialize in satisfying requests rather than providing domain information. For example, decision support agents identify and evaluate options and present recommendations to the user. A workflow agent can be used to ensure that sources are consulted in a preferred, possibly dynamic, order. On-the-fly knowledge building enables users to create *ad hoc* knowledge modules for use in special projects.

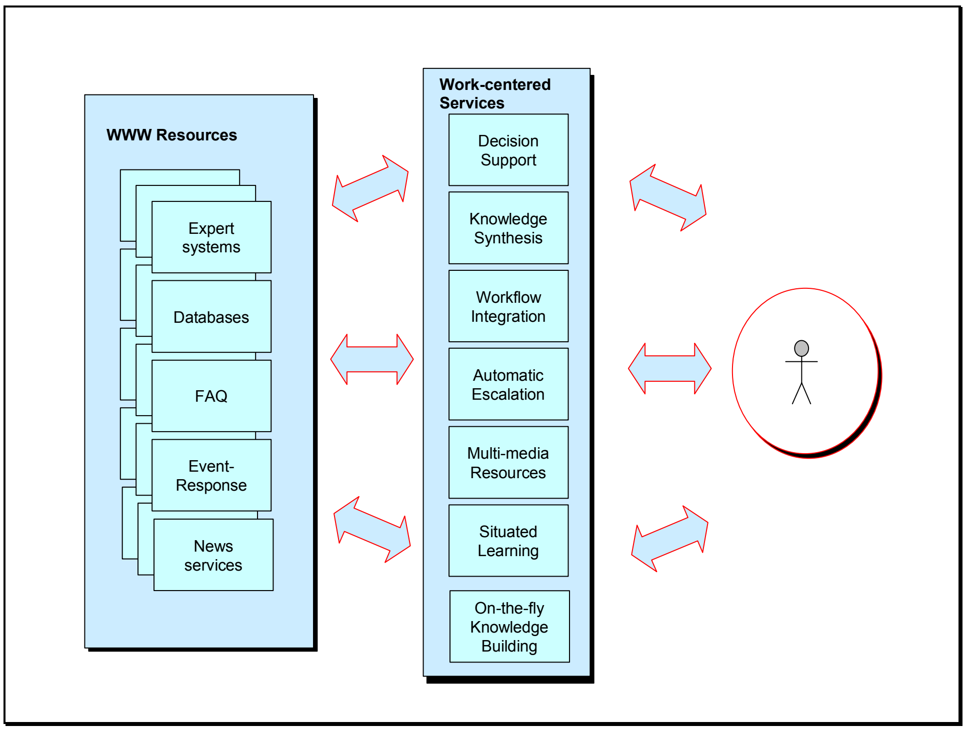

Figure 1 - Work-centered services. Specialized agents functionally interposed between information resources and the user

Underlying Architecture

A key factor in facilitating development of work-centered services is the underlying multi-agent architecture. It must be versatile enough to support a range of agent functionality, and it should centralize complexity that would otherwise be required for individual agent design. These features permit focus on the knowledge-base development, rather than on the details of agent implementation, and provide an effective platform for work-centered service research and development.

The KnoWeb architecture used to implement work-centered services prototype integrates independent resources and engages them in cooperative interaction in goal-driven actions. The language used for inter-agent communication is called KNAML.[2] Based on the conceptual graph linear notation (Sowa), KNAML follows the convention of expressing relations using parentheses, and concepts using brackets, as shown in the example in Figure 2. Unlike conceptual graphs, however, arcs are labeled to facilitate machine understanding. This provides a level of conceptual expressiveness needed to enable Web resources to provide work-centered services, as well as engage in cooperative reasoning.

```
[
 // Registered
 // from service to Sensor reply with null reply to register
 (Register)
          +-anAgent--> [ AgentLocator: ]
          +-aPrecondition--> [ PropositionalFunction: ]
          +-aCapability--> [ PropositionalFunction: [
                [ StartEventStream: ]
          ] ]
          +-anOntology--> [ OntologyLocator: "BlueVelvet" ]
]
```

Figure 2 - KNAML Example. A sensor agent declares its capability to handle StartEventStream requests.

As shown in Figure 3, KnoWeb is powered by a few standard software agents used to perform brokerage, mediation, and conflict resolution. The Meta Agent provides domain-neutral mediation and conflict resolution. By concentrating reusable intelligence in this central resource, redundant complexity in domain and ancillary components is eliminated. The Service Agent provides brokerage services. Work-centered agents and other information resources enroll themselves by registering their capabilities with the Service Agent. Domain Advisor Agents typically help the Meta Agent plan knowledge transactions and manage conflict resolution. The User Agent provides knowledge transformation services between the Meta Agent and the user interface.

The reasoning process employed by the Meta Agent is straightforward. It uses an agenda to keep track of what it is doing and why, and it maintains a context consisting of a collection of propositions asserted in agenda. When the Meta Agent accepts a new request, it first checks to see if the answer is already in the context or if the request is already in its agenda. This enables it to avoid duplicated effort. If the answer is already in the context, the Meta Agent can go ahead and generate a response. Otherwise, it proceeds with problem solving, as depicted in Figure 4.

[2] Obtusely, the acronym suggests "KNAML is not a markup language."

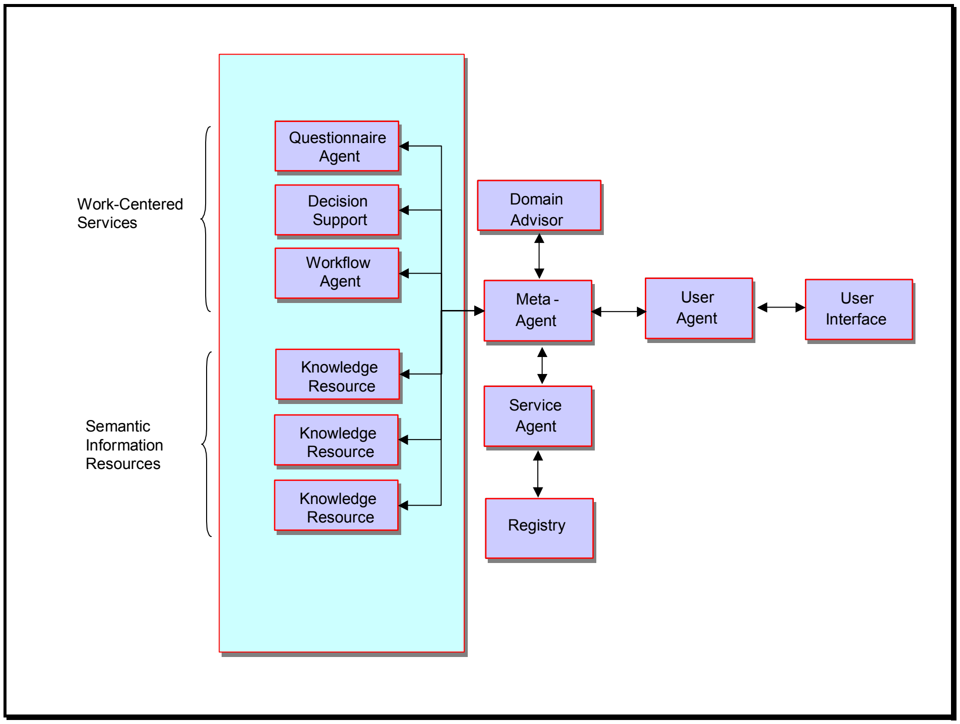

Figure 3 - Example KnoWeb Configuration. Sample work-centered services and knowledge resources

Because most of the heavy lifting required for multi-agent intelligence is performed by the Meta Agent, the design of knowledge and work-centered agents is simple. Agents are implemented using agent wrappers. An agent wrapper is a reusable agent object that can be instantiated to perform a repertoire of functions. Work-centered agents are implemented using wrappers just as are database, symbolic logic, and FAQ agents.

Among the work-centered resources currently prototyped are decision-tree agents, questionnaire agents, and interview agents. Questionnaire agents satisfy their goals by obtaining responses to a battery of questions, as in a survey or questionnaire. Decision-tree agents seek to successfully navigate a decision tree.

The interview agent is a state-based agent responsible for carrying a transaction through a definable workflow, utilizing both questionnaire and decision support resources in the process. In our product recommendation prototype, the interview agent is used to aggregate questionnaire and decision-tree requirements into coherent sets. Each set corresponds to an interview state. The agent proceeds from one state to another only when the former state's requirements are fully defined. For example, in a multi-agent system designed to assist the user in selecting network components, in the initial state, the interview agent collects base requirements (e.g. T1, SNMP, both voice and data), and once this is complete, proceeds to optional states, collecting particular requirements for data and then for voice. Only optional states indicated by the results of the base state are entered, so if the user is requesting a data-only system, issues exclusive to voice systems are not even raised. In other words, the workflow is defined dynamically. Upon completion of the optional states, the interview agent then moves to the recommendation state, whereupon recommendations formulated by decision support are presented.

It is instructive to contrast this scenario to what would transpire were there no work-centered resources available. At one extreme, it would be necessary to handcraft the user interface to ensure that interactions were carried out in a coherent manner. This is a difficult approach in a multi-agent system, where resources have a tendency to be transient and redundant and events are asynchronous. Escalating this issue to the level of the Semantic Web suggests that creating a handcrafted interface would be daunting indeed! Alternatively, expert agents may be left to dispatch questions as they arise within the course of their internal inferencing processes, with utter disregard for the user experience. In this case, requirements are served to the user interface on a first-come-first serve basis, originating from any and all agents with any interest in the matter, irrespective of coherency, redundancy, or timing. Clearly, work-centered workflow management has a place on the Semantic Web.

It may be arguable as to whether these work-centered agents can be fairly construed as knowledge resources. From the agent's perspective, they look like knowledge and they act like knowledge. Implementation of new questionnaire and decision support agents requires only the creation of interrogative and decision knowledge used by the agent. Creating a workflow for the interview agent is a matter of classifying knowledge. To this end, we have implemented simple knowledge building tools that allow subject-matter experts to create modules without software development.

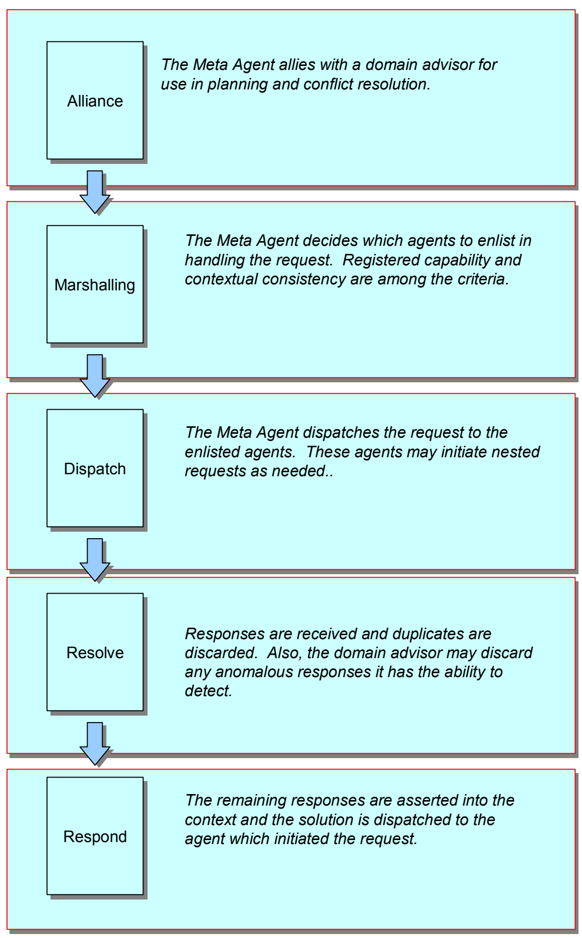

Figure 4 - Meta Agent Reasoning Process.
A simple case

The User Interface

The combined resources of information and work-centered services are manifest in the user interface under auspice of the user agent. The User Agent transforms KNAML specifications into XML documents and dispatches them to the client, where they are used to select, instantiate, schedule, and arrange interactive elements. In the current implementation, these objects are general but simple, consisting of animated graphics, formatted text, sound, and a variety of controls, such as buttons, sliders, and checkboxes. The client application executes in a web-browser. The user interacts by selecting instantiated interaction objects. Completed interactions are dispatched to the user agent from where they are transformed to KNAML and forwarded to the Meta Agent. Figure 5 shows the design of the user interface client as implemented in our prototype event-driven decision support system.

Work-centered services support these activities in two capacities. First, they serve as knowledge resources for determining how information requests may be addressed to the user; second, they provide knowledge-based articulations of solutions reached by other resources. In practice, these capacities are closely intertwined. Only in simplistic models would requests and services be decoupled. Typical activity is progressive. This is reflected both in behind-the-scenes processing and in interactive presentation; that is, at any given time, the interactive context reflects a selective view of the meta agent context, including both explications and user options.

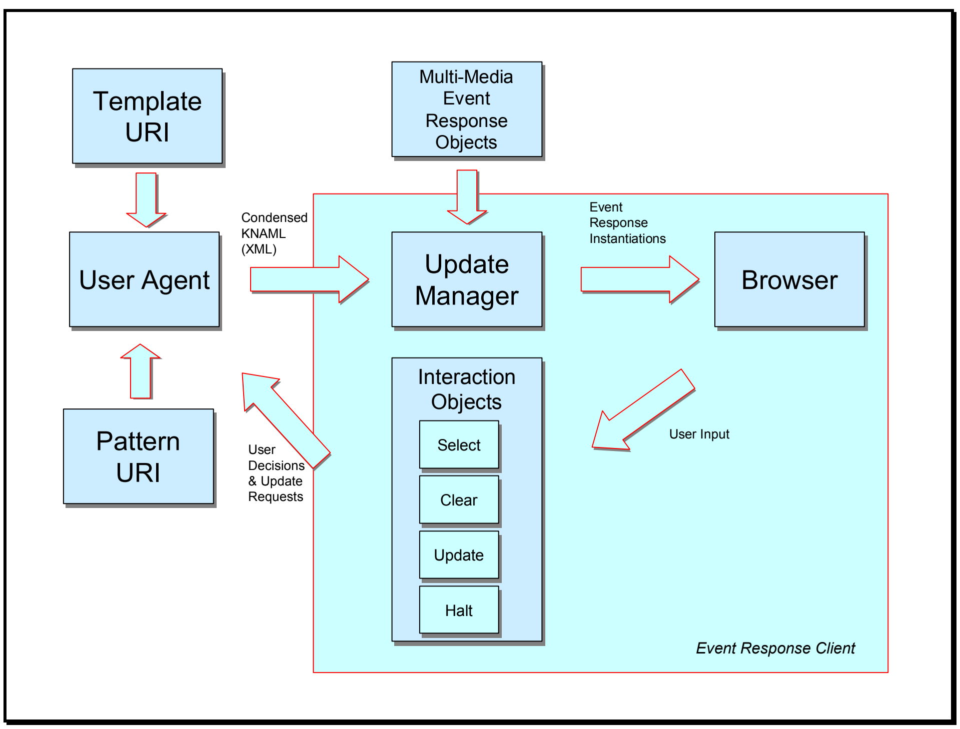

Figure 5 – A Work-Centered Service User Interface Client

In the questionnaire and decision-tree services, user interface modalities are embedded directly in their knowledge modules. This has several advantages. First, it frees the user interface from the constraints of client-side markup languages and facilitates the use of rich media. It also permits user interface logic to be described abstractly during knowledge capture. For example, as recommendations comprising a decision are identified, the types of acceptable options can be specified as well, such as inclusive or exclusive multiple choices, which the user interface renders as checkboxes or radio buttons. By using knowledge representation languages to specify abstract user interface logic, work-centered services can be readily integrated into the Semantic Web.

As the World Wide Web evolves to the Semantic Web, the user interfaces we use will change radically. This is not simply because user interface technology will advance, but because the tasks we use the Web to perform will change. As systems become increasingly ubiquitous, we entrust them with more tasks. As systems become more intelligent, we entrust them with higher levels of responsibility. While entrusting a computer with responsibility might seem farfetched, we already routinely do so in many of our daily activities – shopping, driving, banking, spelling – and we already rely on them for many highly critical tasks, such as fighting wars, flying in space, or monitoring a human heartbeat. This reliance can only be expected to increase in the future. If we are to continue to engage in social relations with these computers, our concept of the user interface must adapt to permit us to speak to them on the same level of intelligence at which they perform.

Conclusions

A variety of services is required to implement the Semantic Web. Among these services will be agents that specialize in supporting people as they go about their tasks. The services these agents provide are called work-centered services. Work-centered services will act as intelligent liaisons between user applications and the greater Web. They will be indirectly accessed by users to formulate and conduct investigations, to collect, organize, and analyze results, and to communicate with users in a humanly coherent manner.

In our work, we have successfully prototyped generalized work-centered services for managing questionnaires and decision support. For both of these services, we have provided knowledge builders to support creation of work-centered knowledge modules. The important point here is not that any innovations in software engineering were required, but that the resulting services can be treated as knowledge resources co-existing with other resources on the Semantic Web.

We have successfully integrated the services into an existing multi-agent system. A key enabling factor here is that the underlying agent architecture is able to eliminate much complexity that would otherwise be required for individual agent design. Further, we have integrated these services with an advanced multi-media user interface. We were able to accomplish this by using work-centered services to abstract the user experience as knowledge and then rendering the knowledge as XML.

Armed with these capabilities, we have been able to implement several prototype products. These include a customer support application, a CRM product recommendation tool, and an event-driven decision support system. Future development will include the definition of additional services, and, on a broader level, an evolving concept of just what constitutes work, as the capabilities of the Web continue to develop. Beauty, perhaps, is only skin-deep, but an effective user interface has depth of character as well as good looks. The architecture of the Web must reflect this depth.

References

Berners-Lee, Tim, James Hendler, and Ora Lassila. "The Semantic Web" *Scientific American.* May, 2001.

Eggleston, R. G., Michael J. Young, Randall Whitaker. "Work-centered Interface Technology: A New Interface Technology for the Battlespace Infosphere." *Nation Aerospace and*

Electronics Conference. October 10-12, 2000. Sinclair Community College, Dayton, Ohio.

Hendler, James and Clayton Lewis. "Introduction: Designing Interfaces for Expert Systems." In *Expert Systems: The User Interface*, Ed. James Hendler. Norwood, NJ: Ablex, 1988.

Lassila, O. and D McGuinness. "The Role of Frame-Based Representation on the Semantic Web." *Knowledge Systems Laboratory Report.* KSL-01-02, Stanford University (2001).

Potter, A. "Direct Manipulation Interfaces." *AI Expert*, 3(10), pp. 28-35, October 1988.

Saravanan, R. "XMLterm: A Mozilla-based Semantic User Interface", http://www.xml.com/pub/a/2000/06/07/xmlterm/ .

Sowa, J. F. *Knowledge Representation; Logical, Philosophical, and Computational Foundation,* Brooks/Cole, Pacific Grove, California 2000

Streeter, G., A. Potter, and T. Flores. "A Mediated Architecture for Multi-Agent Systems," *Workshop on E-Business and the Intelligent Web*, *Seventeenth International Joint Conference on Artificial Intelligence,* Seattle, Washington, August 5, 2001, pp. 173-176.

Streeter, G., and B. Mattern. "KnoWeb: An Agent Architecture for the Semantic Web." *Sentar Technical Reports.* 02-0025. February 15, 2002.