

Intelligent Integration of Information and Services on the Web

Eleni Stroulia

The AAAI 2002 workshop entitled "Intelligent Integration of Information and Services on the Web" was organized by Dean Allemang (Boston University), Eleni Stroulia (University of Alberta), and John Mylopoulos (University of Toronto). It was held on Sunday, 28 July 2002.² The workshop papers are available as a technical report from AAAI Press.

After a welcome and introductions, David Martin of SRI International started the day's presentations with an overview of the DEFENSE ADVANCED RESEARCH PROJECTS AGENCY AGENT MARKUP LANGUAGE (DAML)-based web services description language,³ developed in the context of the Semantic Web Project. The objective of the language is to enable automated software agents to easily accomplish real-world planning tasks by discovering related services, selecting the most appropriate among them, composing them into effective plans, and invoking them to execute these plans and accomplish their tasks.

In the DAML web services language, a service is specified in terms of its profile, its model, and its grounding. The *ServiceProfile* specification is aimed at supporting service discovery and selection. It includes (1) descriptions of the service-defined roles for provider and participants; (2) a functional specification of the service in "planning" terms, that is, input, output, preconditions, and effects; (3) a categorical specification of the service functions in terms of a service taxonomy; and (4) a set of quality at-

tributes. The *ServiceModel* specification is aimed at supporting service invocation, composition, and monitoring and consists of a work-flow model describing how the service is accomplished in terms of atomic and composite processes and their data and control dependencies. Finally, the *ServiceGrounding* specification specifies the implementation-specific details of service invocation, related to protocols, message formatting, and type serialization.

Currently, the group is working on specifying the formal semantics of

■ The evolution of the World Wide Web from a repository of HTML data to a source of varied distributed services creates exciting opportunities for offering complex, integrated services over the web. The syntactic problems of such integration are being addressed by the advent of the web services stack of standards.¹ However, the promise of service integration will not be delivered unless services can be integrated semantically as well. The 2002 AAAI workshop entitled "Intelligent Service Integration" examined this new challenge for the AI community.

the language and the use cases of how the service specifications should be constructed and manipulated. At this point in time, tool support includes specification editors and service matching and brokering. Future work includes developing tool support for planning and run-time execution and monitoring.

Next, Maurine Hatch (University of Alberta) described her work on an infrastructure for specifying reflective agents integrating uniformly wrapped resources to deliver complex services. Agents in the TASK-BASED MEDIATION USING XML (TaMeX) framework are specified in terms of their task structure (a hierarchical work-flow model of their process),⁴ their application-domain model (a model of the concepts they manipulate), and their underlying resources (a set of executable specifications of the resources' functions in terms of the common application-domain model). The TaMeX agents use these declarative specifications at run time to produce an interface for interacting with their end users, control the invocation of the underlying resources and the data flow among the them and the user, and collaborate with each other.

The third presentation of the day, by Hirohisa Naito (Fujitsu Laboratories), described a different approach to integration: Instead of tightly integrating service functions using work-flow specifications, Naito discussed a framework for the opportunistic integration of relevant information in the context of a user's situation. Given a different objective, that is, to use existing web information, the requirements for composition are to track the user's situation and discover opportunities for information use. Consequently, the composition specification language provides primitives for specifying situation information, that is, the conditions under which information is relevant and the relationship among these conditions. The run-time support environment is responsible for efficiently tracking the users' situations and delivering the relevant information to them.

Next, Kiran Mudiam (Arizona State University) presented his work on dynamic component integration based on the specification of their services. This project uses ACME as the representation language for specifying the services provided by the underlying components and provides support for reverse engineering these specifications out of the compo-

nents' command-language syntax.⁵ Furthermore, the associated run-time infrastructure supports the dynamic integration of the specified component services in a client application—including their user interfaces—using JINI.⁶

The final presentation of the morning was by Snehal Thakkar (University of Southern California) on the dynamic composition of geospatial web services. Thakkar described the ARIADNE service-specification language and process and the THESEUS service-execution platform. ARIADNE adopts a relational database view of existing resources and supports the construction of wrappers treating these resources as relational databases. ARIADNE also provides for a translation of the WEB SERVICE DEFINITION LANGUAGE (WSDL) specifications to the ARIADNE service-specification language. Resources specified in ARIADNE can be combined in terms of data-flow specification; these compositions are executed at run time by the THESEUS environment.

In the afternoon, Boi Faltings (Swiss Federal Institute of Technology) presented a survey of five behavioral description languages (PLANNING DOMAIN DEFINITION LANGUAGE [PDDL], DAML- SERVICES [DAML-S], WEB SERVICES FLOW LANGUAGE [WSFL], ELECTRONIC BUSINESS USING XML [EBXML], and CONGOLOG) and comparatively discussed the solutions these languages imply to a set of service-integration challenges: service-integration specification, discovery, and binding of the relevant services, plan composition, and execution. Given that all these languages have their respective advantages, Faltings argued for an abstract behavior representation (ABR) to act as an intermediate representation for exchanging specifications in the underlying languages. Such a representation could also provide the means for characterizing the complexity of particular language combinations and service challenges.

The final presentation of the workshop was given by Ion Muslea (University of Southern California) on wrapper induction. STALKER, the example-based wrapper-induction algorithm developed in the context of

ARIADNE, produces rules for extracting information of interest from HTML documents. Given that the same examples can be described in terms of multiple alternative sets of rules, multiview wrapper-induction algorithms have been developed for the same process. These algorithms are potentially more efficient but fail when the alternative views eventually diverge. To address this problem, Muslea developed an algorithm for evaluating the convergence of multiple views.

Discussion

After the paper presentations, the workshop participants engaged in a discussion on the various themes invoked by the presentations.

The first issue to be addressed was the definition of the term *intelligent* in the context of service integration. The underlying question is essentially, How do the requirements for intelligent service integration go beyond the mechanics of distributing services over the web? The most prominent characteristic of intelligent integration, the participants agreed, is semantics. Work on DAML-S, as well as in projects such as the ones described by Hatch, Mudiam, and Thakkar, aims at providing support for defining the domain-specific semantics of the offered services. Earlier frameworks for distributed component integration on the web have focused on developing a common syntax for specifying components so that they can interact and exchange data but have left the responsibility of constructing meaningful compositions to the developer. The new challenge is to enable automated agents to construct such meaningful compositions. The implication, therefore, is that intelligent service integration frameworks have to provide support for the specification of the semantic functions and constraints of the underlying resources.

However, although the need for semantic specification languages is agreed on, a consensus on the "style" of the specification language does not seem immediate. Three different types of specifications were presented

in the workshop: (1) taxonomic representations of service categories (DAML-S), (2) planning-style specifications of functional input-output specifications and preconditions and postconditions (DAML-S, TAMEX, the system developed by Mudiam et. al, ARIADNE), and (3) procedural specifications of applicability conditions (refer to the presentation by Naito). These different specification styles are aimed at different types of integration problems, and translation mechanisms among them, such as the ABR proposed by Faltings, could be the future challenge. Furthermore, the computational complexity of matching can be expensive. The intelligent integration research effort seems to be faced with a strategic decision about whether to require the expensive formal specifications or whether to allow for less costly, heuristic semi-formal approaches.

Irrespective of the specification style, it is clear that to have an impact, any service-specification language should provide methods and tool support for constructing, publishing, selecting, composing, and executing specifications and their compositions. The various projects represented in the workshop have organized these subproblems in different priority order. For example, work on DAML-S has focused on developing the formal semantics of the underlying specification language and producing tools to support the editing, publication, and selection of specifications. The TAMEX and service-based component integration projects have placed more emphasis on the problem of reverse-engineering specifications out of existing (possibly legacy) components and on support for the flexible run-time execution of the composite services, including the invocation of the underlying components and their interaction with the end user. Finally, the ARIADNE and THESEUS framework provides support for learning specifications out of user-annotated examples of existing web documents, composing the specifications in a query-planning manner and executing the compositions at run time.

The role of the end user for the

specified services and their compositions is also not commonly agreed on. The DAML-S effort seems to assume, at least initially, that the end user of the semantic services infrastructure will be the developer of the compositions. In contrast, the work in TAMEX, the ACME-based component integration project and the Fujitsu project, views as the end consumer the user of the composite application and is also concerned with the interface through which the user should be able to issue requests and receive responses from the composite application.

Finally, an issue that will probably be receiving increasing attention in the future is the implication of contractual agreements regarding web services. As the development of web-based applications starts to rely increasingly on third-party services, a framework for providing, evaluating, and paying for quality-assurance specifications of these services will become necessary. The DAML-S language has already provided for the description of attributes such as capabilities and functional attributes in the service profile. However, if the consumers were to use services for developing new applications, a mechanism for verifying the stated quality attributes would be required. The implication is that these attributes would have to be quantifiable to some degree, which, in turn, implies specific requirements on the expressiveness of this aspect of the language. Furthermore, one can envision a cost structure reflecting the specified quality attributes and "discount factors" to deal with potential failures of the provider to meet these attributes. However, it is important to note that many of the underlying implementation platforms (SIMPLE OBJECT ACCESS PROTOCOL [SOAP], for example) have not yet addressed more basic quality issues such as security and transactional integrity. Addressing these issues will be of critical importance in ensuring that the intelligent integration technologies have an impact.

Notes

1. www.webservices.org/index.php/article/archive/3/.

2. The schedule can be found at www.cs.ualberta.ca/~stroulia/AAIWISI/schedule.-html.
3. www.daml.org/services.
4. www.cs.ualberta.ca/~stroulia/TAMEX.
5. www-2.cs.cmu.edu/~acme.
6. www.sun.com/software/jini.

Eleni Stroulia obtained her B.Sc. in computer engineering from the University of Patras, Greece, and her M.Sc. and Ph.D from the College of Computing at the Georgia Institute of Technology. She is now an assistant professor in the Comput-



er Science Department at the University of Alberta. Her research focuses on applying AI methods to address software engineering problems. More specifically, her team has been using planning and learning methods to model legacy application user interfaces and induce wrappers for making their services accessible from various web platforms. Currently, her team is working on developing intelligent-agent support for web-service reengineering, discovery, adaptation, and composition. Her e-mail address is stroulia@cs.ualberta.ca.



Simulating Organizations

Computational Models of Institutions and Groups

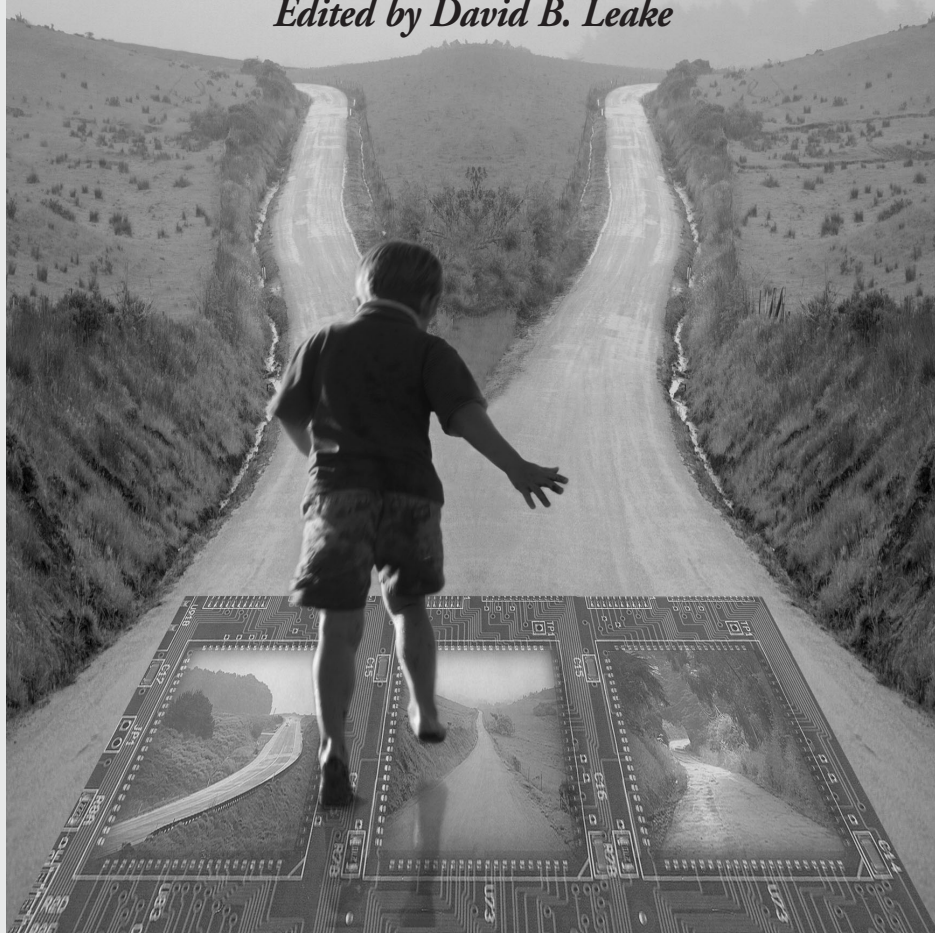
Edited by Michael J. Prietula, Kathleen M. Carley, and Les Gasser

6 x 9, 350 pp., \$45.00, ISBN 0-262-66108-X (Prices higher outside the U.S. and subject to change without notice.)
To order, call 800-356-0343 (US and Canada) or (617) 625-8569.
Distributed by The MIT Press, 5 Cambridge Center, Cambridge, MA 02142

Case-Based Reasoning

Experiences, Lessons, & Future Directions

Edited by David B. Leake



Case-based reasoning is a flourishing paradigm for reasoning and learning in artificial intelligence, with major research efforts and burgeoning applications extending the frontiers of the field. This book provides an introduction for students as well as an up-to-date overview for experienced researchers and practitioners. It examines the field in a “case-based” way, through concrete examples of how key issues — including indexing and retrieval, case adaptation, evaluation, and application of CBR methods — are being addressed in the context of a range of tasks and domains. Complementing these case studies are commentaries by leading researchers on the lessons learned from experiences with CBR and visions for the roles in which case-based reasoning can have the greatest impact. (ISBN 0-262-52110-x) 420 pp., index, bibliography.

The AAAI Press • Distributed by The MIT Press
Massachusetts Institute of Technology, Cambridge, Massachusetts 02142
To order, call toll free: (800) 405-1619 or go to <http://mitpress.mit.edu>
MasterCard and VISA accepted. <http://www.aaai.org>