

WRITTEN BY **RAHUL SINGH, KATIA SYCARA, TERRY PAYNE**

Distributed AI, Schedules, and the Semantic Web

RCal provides a glimpse of what's to come

AUTHOR BIOS

Rahul Singh is a graduate student at the Robotics Institute at Carnegie Mellon University and also works as a research programmer with the Advanced Agent Technology Laboratory at Carnegie Mellon University. He received his B. Eng. in computer systems engineering from the University of South Australia, Adelaide, and is currently pursuing a master's degree in robotics.

Katia Sycara is a research professor at the Robotics Institute at Carnegie Mellon University and director of the Advanced Agent Technology Laboratory. She received a BS in applied mathematics from Brown University, an MS in electrical engineering from the University of Wisconsin, and a PhD in computer science from the Georgia Institute of Technology. She is the founding editor-in-chief of Autonomous Agents and Multi-Agent Systems.

Terry R. Payne is a lecturer at the University of Southampton, UK. He received a B.Sc. in computer systems engineering from the University of Kent at Canterbury, UK, and his M.Sc. and PhD in artificial intelligence from the University of Aberdeen, Scotland. He is currently involved in the DAML program, and is a coauthor of the DAML-S Service Description Language.

Meetings are an integral part of modern life, regardless of whether they're formally established at the workplace or casual agreements made over the phone. Many factors affect meeting scheduling, some of which are explicit (What existing meetings do I have?), implicit (I prefer to avoid meetings before 10:00 am), or cultural (social events should be scheduled for Friday evenings and weekends, but not Sunday mornings).

In addition, humans typically reason about where the meeting will take place, its duration, purpose, and so on. All these factors contribute to making meeting scheduling a difficult problem. However, recent developments in automated preference acquisition, multiagent negotiation, and reasoning techniques for semantic content on the Web are slowly making automated scheduling a reality.

An important factor in automated meeting scheduling is being able to share, understand, and reason about all the details of an event or meeting request. The Semantic Web facilitates the representation and distribution of knowledge as structured data with meaning, thus allowing agents to reason about concepts in the real world. We've developed several intelligent agents that negotiate with each other to organize meetings on behalf of their users, using published knowledge to make appropriate decisions. The RETSINA Semantic Web Calendar Agent (RCal) is one such agent built using the RETSINA AI infrastructure, which augments a widely used Personal Information Manager (PIM) – MS Outlook 2000. RCal combines knowledge about its user's current schedule, information about colleagues and friends (using MS Outlook 2000's Contact entries), and knowledge gathered from the Semantic Web to better automate meeting scheduling and management. In this scenario, users running MS Outlook 2000 on their desktops also have an instance of RCal running in the background, acting on their behalf.

RCal (see Figure 1) schedules meetings for its user, updates the user's calendar with schedules from the Semantic Web, interacts with Web services that may provide additional relevant information pertaining to scheduled meetings, and provides alerts based on occurring events.

Distributed Meeting Scheduling

RCal negotiates with other agents to find mutually agreeable times based on the user's availability and preferences. Traditionally, the burden of maintaining an up-to-date calendar has fallen on the user – a task that is time consuming and error prone. To address this, RCal can reason about events and schedules published on the Semantic Web, and automatically incorporate them directly into the user's schedule. This reduces the burden on the user, and maintains an up-to-date calendar that can be consulted by the agent when scheduling meetings.

RCal currently supports two types of distributed meeting negotiation – multiparty negotiation and appointment-request negotiation. Multiparty negotiation occurs when several agents try to identify a mutually agreeable meeting slot based on their users' current schedules and preferences, whereas appointment-request negotiation identifies possible meeting times for one party based on a meeting request. This latter form of negotiation is used by the Web-based E-Secretary to allow people to request meetings or appointments via a Web-based interface.

RCal's multiparty negotiation occurs when someone desires a meeting with one or more individuals, each of which employ their own RCal agents to manage their schedules. RCal goes through several rounds of automated negotia-

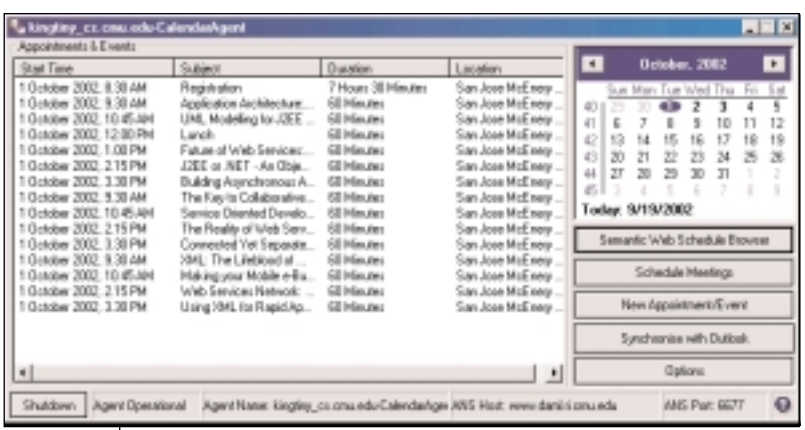

FIGURE 1 RETSINA Semantic Web Calendar Agent

tion with other RCal agents until all the agents agree upon a common time. Users can instruct RCal to schedule a meeting by specifying a particular time they would prefer or a window within which an appropriate slot of the specified duration should be found. For example, a user may request "an hour-long meeting with all the members of the project group within the next two days." The negotiation uses the Contract Net Protocol, which starts with an initial document (a contract) specifying details of the event, such as its attributes (subject, body location, etc.) and its constraints (start time, end time, and duration). The contract is broadcast to the other RCal agents, which consult their users' calendars and reply with one of three types of bid:

- Accept the contract for the meeting at the specified time
- Reject the contract outright (i.e., do not wish to meet)
- Reject the time specified by the contract and propose alternative time slots

The host evaluates all bids upon receipt and looks for a time that is acceptable to all the attendees. If a common time is found, the host sends out awards corresponding to the received bids and the RCal agents update their calendars before sending out confirmations that terminate the negotiation. If no common time can be found, the negotiation restarts with a new contract generated within the original constraints specified by the host. This iteration continues until a meeting time is identified or no new contract can be generated due to the original constraints set by the host of the meeting.

The RETSINA E-Secretary

There may be occasions when users don't have access to their RCal agent, or may not use a PIM to manage their calendar and hence need some alternative mechanism for requesting meetings. Alternatively, individuals or organizations (such as clinics or dental offices) may want to publish a Web-based interface through which appointments can be requested. The RETSINA E-Secretary agent (see Figure 2) is a Web-based agent that facilitates appointment requests, without the need for both parties to use RCal. The design is based on the concept of scheduling meetings via a human secretary – the secretary interacts with meeting requesters and manages meeting requests according to when time slots are available. In such cases, the negotiation is limited to only two parties: a human requesting a meeting or appointment at a preferred time, and RCal responding to this request (via the E-Secretary) with the appropriate meeting time based on the requestee's calendar. Typically, users who manage their calendars using RCal also have E-Secretary agents running on their behalf on the Web.

Anyone desiring a meeting enters details such as name, e-mail address, location for the meeting, and desired time via a form presented by the E-Secretary. The E-Secretary then sends the meeting request to RCal, which looks for an appropriate time slot. If one is found, it's presented to the human requesting the meeting, who can either accept or reject the proposed meeting time. If the meeting is accepted, the E-Secretary sends a confirmation to RCal, which updates the calendar and notifies both parties of the scheduled meeting via e-mail.

Maintaining Calendars

One advantage that RCal has over many other agent-based meeting scheduling systems is the ability to gather relevant information from the Semantic Web. Traditionally, calendar managers relied on humans to enter meetings that were not automatically entered. While this approach works for occasional events, it breaks down when large-scale schedules (such as conference schedules) need to be added. In many cases, users simply enter single events to represent the whole schedule, and mark the time as busy, an approach that prohibits further negotiation during this time. For example, "Bob" might plan to attend the three-day Web Services Edge Conference, and hence would want to update his calendar to reflect this. If this conference is entered as a single event, RCal won't schedule any meetings during this time. However, this doesn't accurately reflect Bob's actual schedule, since it doesn't take into account coffee and lunch breaks, and talks or presentations he may choose not to attend. In addition, Bob won't benefit from being able to consult his calendar to find out when individual events occur, or get reminders sent to his PDA or mobile phone. More important, it's often desirable to schedule meetings with other delegates at a conference, yet at such events, access to PIMs, schedules, and the delegates themselves can be difficult.

RCal and the E-Secretary overcome this problem by importing schedules directly from the Semantic Web. Traditionally, extracting schedules from the World Wide Web has been a problem since HTML (currently used to publish schedules) requires custom-built software tools, such as screen scrapers, to elicit the relevant information from the Web pages. Although XML representations can be used to simplify this, such an approach requires the standardized use of a single DTD or XML Schema. The Semantic Web relaxes this constraint by providing a framework (built upon XML) within which ontologies (formal specifications on how to represent concepts) can be built to describe concepts in the real world. Additionally, AI-based reasoning tech-

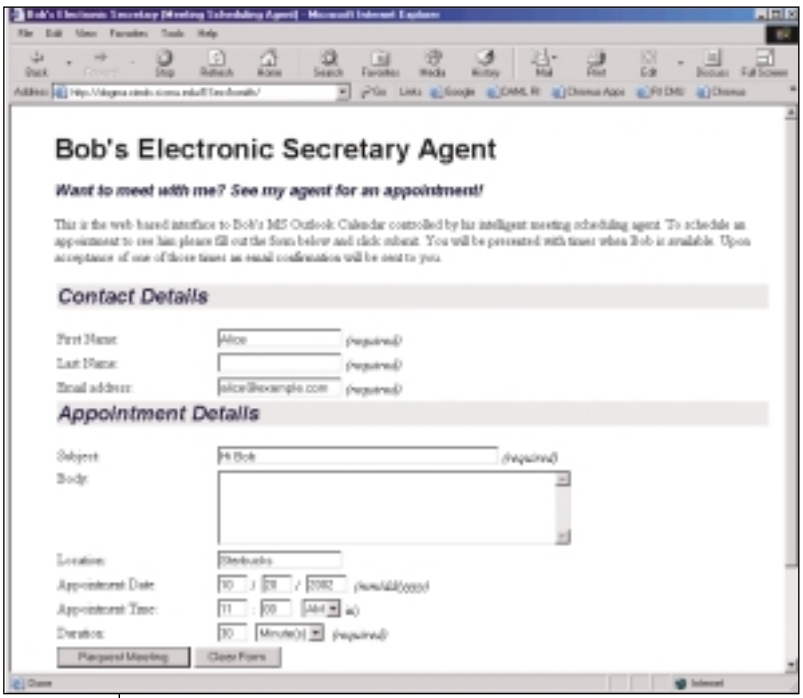

FIGURE 2 Alice schedules a meeting with Bob through his E-Secretary agent

niques can be used within Web services to search, translate, merge, or navigate across markup in these ontologies.

The Semantic Web initiative utilizes RDF to mark up knowledge and publish it to the World Wide Web, where software agents can access it. Moreover, information can be made available in a structured form with links to other pieces of information, much in the way Web pages are currently linked together. The Hybrid iCal ontology, derived from the iCalendar specification, is one of several ontologies that provide a framework for schedules to be marked up in RDF and published on the Web. Used by several applications (including RCal), it allows sharing and reasoning of schedules. Listings 1-3 illustrate a schedule marked up in RDF (the namespace declaration and <rdf:RDF> tags have been removed for brevity) with three events for the Web Services Edge Conference.

In this example the iCal onthology is used to present events in the schedule, whereas the Dublin Core ontology (xmlns:dc) is used to mark up meta-information about the document itself, such as the source, title, and description. Fields in RDF can either be populated with simple text strings or with references to URIs (Uniform Resource Identifiers) that point to other RDF concepts. A URI reference such as this allows for more information to be extracted about the concept in question, which could be located elsewhere on the Web. This framework leads directly from the present architecture of the World Wide Web, except that now it links knowledge rather than plain text.

A schedule may contain multiple calendars (or schedules) represented by instances of the RDF class <ical:VCALENDAR>. Each calendar contains properties (<ical:VEVENT-PROP>) that link multiple events (<ical:VEVENT>) in the calendar. The schedule in Listing 1 contains four events of which the "Registration" event is inline, while the others reference a URI to resources in another document (e.g., the http://www.daml.ri.cmu.edu/Schedules/WSE2002-JavaTrack.rdf#J1). Each event in the file WSE2002-JavaTrack.rdf contains information about the event, such as its start time, duration, etc., exactly in the way the "Registration" event is marked up. Here the events related to each track of the conference are in another document, and the URIs allow an agent to navigate across this web of knowledge and extract information about them.

The formats for the fields (such as <ical:DATE-TIME> and <ical:DURATION>) depend upon the designer of the ontology and could very well be explicitly broken up into individual fields of day, month, year, and so on, but comprehensive markup should reference URIs to resources that represent time using an ontology that allows temporal reasoning.

RCal can import a schedule such as this using the Semantic Web Schedule Browser (see Figure 3) and present the information to the user in an organized manner, allowing more information to be retrieved by right-clicking on the concepts in question. Events can also be selected and imported into Outlook, thus allowing a user to update his/her calendar without having to type out the details of each event.

Ontologies: The Building Blocks

Marking up schedules in RDF ensures that they can be shared between different applications without assuming a tightly defined standard, thus allowing calendars to be kept up to date. An application that understands schedules defined by one ontology may be able to reason about a schedule defined by another ontology through AI-based reasoning techniques and articulations (rules that map concepts from one ontology to another). This is useful, as new ontologies are continually being developed on the Semantic Web. However, on-the-fly resolution of semantic mismatches may not be possible. In such situations, it may be desirable to delegate the task of translating a schedule from an unknown representation into one that is familiar, another agent or service provider. The DMA2ICal Service is one such service-oriented application that can be located through a semantic-based discovery service.

The DMA2ICal Service converts markup based on the DAML Meeting Agenda (DMA) ontology to that based on the Hybrid ICal ontology, and demonstrates several technologies:

- How simple translation services can convert markup based on one ontology to that of another ontology
- How agents can utilize translation-based Web services when encountering unknown markup
- How agents can utilize the semantic-based DAML-S discovery service

When RCal encounters schedules marked up based on an unknown ontology, it attempts to identify the top-level concepts, and uses these to generate service requests that can be submitted to a discovery service. Though infrastructures for discovery, such as UDDI, are slowly being deployed, they typically provide white-page (name lookup), or yellow-page (capability-based) lookup. The DAML-S Matchmaker is a lookup service that uses a DAML (DARPA Agent Markup Language)-based logical reasoning engine and utilizes resources on the Semantic Web. DAML-S is the DARPA Agent Markup Language for Services ontology, which provides a framework and a set of resources for performing semantic-based service discovery.

The DAML-S Matchmaker attempts to match the request submitted by RCal with previously advertised capability descriptions, and returns a list of the names of agents (or services) that provide the desired service. The DAML-S service profile that advertises the DMA2ICal translation service is shown in Listing 4. This profile represents the service named "DMA2ICal," which takes a "Meeting" object as input and returns a "VCALENDAR" object as output. The <profile:restrictedTo> tags indicate that the input (Meeting) and output (VCALENDAR) objects lie in the domain of the ontology referred to. A profile such as this describes a service in details that allow an agent to reason about concepts such as the Meeting concept in the domain of events that occur in time rather than, say, a meeting in the sense of a merger of two physical objects.

Listing 5 shows one of the events of the Web Services Edge Conference in the DMA ontology, and Listing 6 shows its corresponding translated schedule document in the iCal ontology. The DMA ontology contains concepts analogous to those within the iCal ontology, although the ontologies themselves aren't logically equivalent. While this lack of logical equivalence means that some schedules defined in one ontology may not be representable in another, there is a subset of schedules that can be represented by both ontologies. It's therefore possible to define an agent or service that algorithmically translates between the two representations, and once advertised, any agent may be

"Marking up schedules in RDF ensures that they can be shared between different applications without assuming a tightly defined standard"

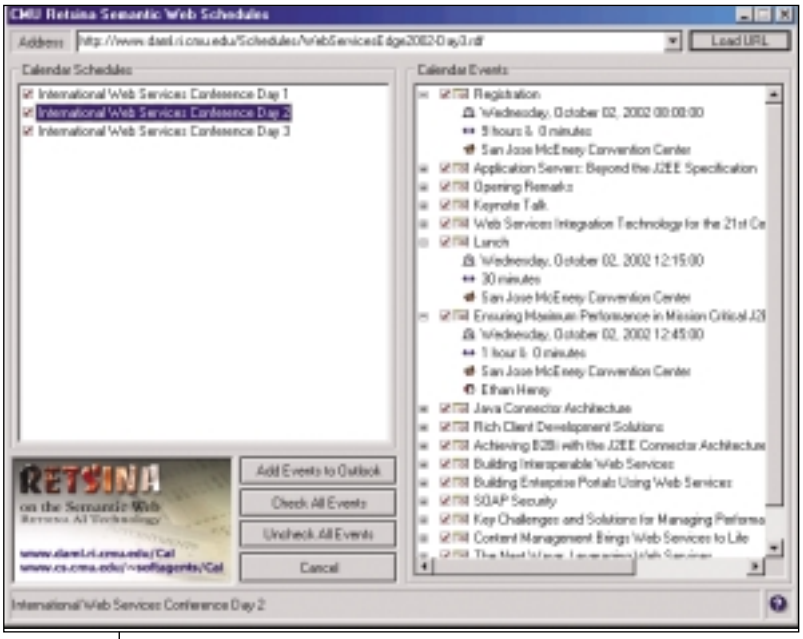

FIGURE 3 Schedules in the Semantic Web Schedule Browser

requested to perform this task. For example, the DMA2ICal translation service ensures that the start time (represented by a "DataType" entry such as a string, and linked to the <dma:Day> concept by the property <dma:time>) is translated into the concept <ical:DATE-TIME>, the value of which is equivalent (although formatted differently) to the DataType string.

Future Work

The Semantic Web allows RCal to gather and reason about information in the scheduling domain, and use this to facilitate automated distributed meeting scheduling. By adding Web-based and PDA-based agents, access to and management of one's calendar in a ubiquitous environment is becoming a reality. The Semantic Web aims to publish knowledge that will be machine understandable. Given this, distributed communities of agents and Web services will collaborate to provide intelligent assistance to the user, replacing the need for single, monolithic systems. These communities, driven by a distributed knowledge base such as the Semantic Web, will give us access to information currently unattainable without tedious searches and manual navigation.

To achieve these goals, improved methods for service discovery and negotiation will be required. The DAML-S initiative is currently developing ontologies and tools to facilitate service discovery. However, more work is required to explore issues related to semantic interoperability within service communication. The current version of RCal has prior knowledge of the DMA2ICal translation service – once the service has been located, it is invoked as a simple cgi-bin script. Further work is required to extend it to fully exploit the DAML-S interface definition to relax assumptions on message formats for invocation.

Another problem is that of generally describing the capabilities of services semantically, as in the case of the DMA2 ICal service with the DAML-S capability description language. The DMA2ICal service here is relatively simple, but more complex services involving many interactions with other supporting services will require more complex descriptions that will then need to be found and reasoned about. The development of a framework to allow such reasoning in open environments is currently under way.

As a first step however, RCal gives us a glimpse of things to come, as intelligent agents begin to navigate the large knowledge base the Web promises to evolve into. RCal is one of the first killer applications that demonstrate the utility of the Semantic Web, using knowledge to provide assistance at a level higher than that available from current applications.

References

- Smith, R.G. "The Contract Net Protocol: High-Level Communications and Control in a Distributed Problem Solver." *IEEE Transactions on Computers*, December 1980.
- Hendler, J. and McGuinness, D.L., "DARPA Agent Markup Language." IEEE Intelligent Systems. Vol. 15, issue 6.
- *Resource Description Framework (RDF) Model and Syntax Specification*: www.w3.org/TR/REC-rdf-syntax
- Ankolekar, A., Burstein, M., Hobbs, J., Lassila, O., Martin, D., McIlraith, S., Narayanan, S., Paolucci, M., Payne, T., Sycara, K., and Zeng, H. "DAML-S: Semantic Markup for Web Services." International Semantic Web Working Symposium, 2001.
- Ankolekar, A., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., Martin, D., McIlraith, S., Narayanan, Paolucci, M., Payne, T., and Sycara, K. "DAML-S: Web Service Description for the Semantic Web." Proceedings of the 1st International Semantic Web Conference (ISWC), 2002.
- Paolucci, M., Kawamura, T., Payne, T., and Sycara, K. "Semantic Matching of Web Services Capabilities." First International Semantic Web Conference, 2002.
- Hendler, J. "Agents and the Semantic Web."*IEEE Intelligent Systems*. March/April 2001.
- Berners-Lee, T., Hendler, J., and Lassila, O. "The Semantic Web." *Scientific American*. May 2001.
- Sycara, K., Paolucci, M., Van Velsen, M., and Giampapa, J.A., "The RETSINA MAS Infrastructure" technical report CMU-RI-TR-01-05. Robotics Institute, Carnegie Mellon University. March 2001.
- *The iCalendar Specification*: www.ietf.org/rfc/rfc2445.txt
- Payne, T.R., Singh, R., and Sycara, K., "Calendar Agents on the Semantic Web." *IEEE Intelligent Systems*. IEEE Computer Society. May/June 2002.
- *The Resource Description Framework (RDF)*: www.w3c.org/RDF
- *The UDDI Technical white paper*: www.uddi.org/whitepapers.html
- Payne, T.R., Singh, R., and Sycara, K., "RCAL: A Case Study on Semantic Web Agents," First International Conference on Autonomous Agents and Multiagent Systems, 2002.

KINGTINY@CS.CMU.EDU

KATIA@CS.CMU.EDU

TRP@ECS.SOTON.AC.UK

LISTING 1

```
<!-WebServicesEdge2002-Day1.rdf -->
<ical:VCALENDAR
rdf:ID="WebServicesEdge2002-Day1">
 <dc:source rdf:resource="http://www.sys-
con.com/WebServicesEdge2002West/sched.cfm"
/>
  <dc:description>International Web
Services Conference Day 1</dc:description>
 <dc:contributor>
  <foaf:Person>
    <foaf:name>Rahul Singh</foaf:name>
     <foaf:mbox
rdf:resource="mailto:kingtiny@cs.cmu.edu"
/>
  </foaf:Person>
 </dc:contributor>
  <dc:date>2002-08-25</dc:date>
 <ical:VEVENT-PROP>
  <ical:VEVENT>
   <ical:DTSTART>
    <ical:DATE-TIME>
       <ical:TZID rdf:resource="#PDT" />

<rdf:value>20021001T083000</rdf:value>
     </ical:DATE-TIME>
    </ical:DTSTART>
   <ical:DTEND>
    <ical:DATE-TIME>
       <ical:TZID rdf:resource="#PDT" />

<rdf:value>20021001T160000</rdf:value>
     </ical:DATE-TIME>
    </ical:DTEND>
    <ical:LOCATION
rdf:resource="#SJMcEneryConvCenter" />

<ical:DESCRIPTION>Registration</ical:DESCR
IPTION>
   </ical:VEVENT>
  </ical:VEVENT-PROP>
  <ical:VEVENT-PROP
rdf:resource="http://www.daml.ri.cmu.edu/S
chedules/WSE2002-JavaTrack.rdf#J1" />
  <ical:VEVENT-PROP
rdf:resource="http://www.daml.ri.cmu.edu/S
chedules/WSE2002-JavaTrack.rdf#J2" />
</ical:VCALENDAR>
```

LISTING 2

```
 <!-WSE2002-JavaTrack.rdf -->
<ical:VEVENT rdf:ID="J2">
 <ical:DTSTART>
  <ical:DATE-TIME>
   <ical:TZID rdf:resource="#PDT" />
   <rdf:value>20021001T104500</rdf:value>
  </ical:DATE-TIME>
 </ical:DTSTART>
 <ical:DTEND>
  <ical:DATE-TIME>
   <ical:TZID rdf:resource="#PDT" />
   <rdf:value>20021001T114500</rdf:value>
  </ical:DATE-TIME>
 </ical:DTEND>
 <ical:LOCATION rdf:resource="
#SJMcEneryConvCenter" />
 <ical:DESCRIPTION>
 UML Modelling for J2EE and Web Services
 </ical:DESCRIPTION>
 <ical:ATTENDEE>
  <ical:CAL-ADDRESS>
   <rdf:type rdf:resource="foaf:Person" />
   <foaf:name>Oliver Le
Diouris</foaf:name>
   <foaf:fundedBy>Oracle</foaf:fundedBy>
  </ical:CAL-ADDRESS>
 </ical:ATTENDEE>
</ical:VEVENT>
```

LISTING 3

```
<!-WSE2002-JavaTrack.rdf -->
<ical:VEVENT rdf:ID="J1">
 <ical:DTSTART>
  <ical:DATE-TIME>
   <ical:TZID rdf:resource="#PDT" />
   <rdf:value>20021001T093000</rdf:value>
  </ical:DATE-TIME>
 </ical:DTSTART>
 <ical:DTEND>
  <ical:DATE-TIME>
   <ical:TZID rdf:resource="#PDT" />
   <rdf:value>20021001T103000</rdf:value>
  </ical:DATE-TIME>
 </ical:DTEND>
 <ical:LOCATION
rdf:resource="http://www.daml.ri.cmu.edu/S
chedules/WebServicesEdge2002-
Day1.rdf#SJMcEneryConvCenter" />
 <ical:DESCRIPTION>
 Application Architecture: A necessary
foundation for WebServices
 </ical:DESCRIPTION>
 <ical:ATTENDEE>
  <ical:CAL-ADDRESS>
   <rdf:type rdf:resource="foaf:Person" />
   <foaf:name>Walter Hurst</foaf:name>
   <foaf:fundedBy>Wakesoft</foaf:fundedBy>
  </ical:CAL-ADDRESS>
 </ical:ATTENDEE>
</ical:VEVENT>
```

LISTING 4

```
translation service
<service:serviceProfile
rdf:ID="ontology_translation_service" />
<profile:Profile>

<profile:serviceName>DMA2ICAL</profile:ser
viceName>
 <profile:textDescription>Service trans-
lates schedules in RDF from the     iCal
ontology to the DMA Ontology
 </profile:textDescription>
 <profile:contactInformation>
  <profile:Actor
rdf:resource="http://www.cs.cmu.edu/~kingt
iny/
kingtiny.rdf" />
 </profile:contactInformation>
 <profile:input>
  <profile:ParameterDescription
rdf:ID="DMA_Meeting_Object">
   <profile:parameterName>Meeting</pro-
file:parameterName>
   <profile:restrictedTo
rdf:resource="http://http://www.daml.org/
2001/10/agenda/agenda-ont.daml#Meeting" />
  </profile:ParameterDescription>
 </profile:input>
 <profile:output>
  <profile:ParameterDescription
rdf:ID="iCal_VCALENDAR_Object">

<profile:parameterName>vcal</profile:param
eterName>
   <profile:restrictedTo
rdf:resource="http://ilrt.org/
     discovery/2001/06/schemas/ical-
full/hybrid.rdf#VCALENDAR" />
  </profile:ParameterDescription>
 </profile:output>
</profile:Profile>
```

L isting5:

LISTING 5

```
<!—WSESchedule-DMA.rdf -->
<dma:Meeting ID="WebServicesEdge2002-DMA">
 <dma:name>
  Web Services Edge 2002 West
 </dma:name>
 <dma:location>
  San Jose McEnery Convention Center
 </dma:location>
 <dma:day>
<dma:Day>
 <dma:start>
  2002-10-01TO09:30:00
 </dma:start>
  <dma:items rdf:parseType="daml:collec-
tion">
   <dma:Talk>
    <dma:title>
     Application Architecture: A Necessary
Foundation for Web Services
    </dma:title>
    <dma:speaker>
     <dma:Speaker rdf:ID="baron">
      <dma:name>
       Walter Hurst
      </dma:name>
      <dma:organisation>
       Wakesoft
      </dma:organisation>
     </dma:Speaker>
    </dma:speaker>
    <dma:duration>
     PT60M
    </dma:duration>
   </dma:Talk>
  </dma:items>
</dma:Day>
</dma:day>
</dma:Meeting>
```

LISTING 6

```
<!—WSE2002-JavaTrack.rdf -->
<ical:VEVENT rdf:ID="J1">
 <ical:DTSTART>
  <ical:DATE-TIME>
   <ical:TZID rdf:resource="#PDT" />
   <rdf:value>20021001T093000</rdf:value>
  </ical:DATE-TIME>
 </ical:DTSTART>
 <ical:DTEND>
  <ical:DATE-TIME>
   <ical:TZID rdf:resource="#PDT" />
   <rdf:value>20021001T103000</rdf:value>
  </ical:DATE-TIME>
 </ical:DTEND>
 <ical:LOCATION>
  <ical:GEO rdf:ID="SJMcEneryConvCenter">
   <ical:GEO-NAME>
    <ical:TEXT
     rdf:value="San Jose McEnery
Convention Center" />
     </ical:GEO-NAME>
    </ical:GEO>
 </ical:LOCATION>
 <ical:DESCRIPTION>
 Application Architecture: A necessary
foundation for WebServices
 </ical:DESCRIPTION>
 <ical:ATTENDEE>
  <ical:CAL-ADDRESS>
   <rdf:type rdf:resource="foaf:Person" />
   <foaf:name>Walter Hurst</foaf:name>
   <foaf:fundedBy>Wakesoft</foaf:fundedBy>
  </ical:CAL-ADDRESS>
 </ical:ATTENDEE>
</ical:VEVENT>
```

Download the Code
www.sys-con.com/xml