# Semantic Web Services as Agent Behaviors

Paul A. Buhler[1], José M.Vidal[2]

[1] College of Charleston, Dept. of Computer Science,
66 George Street, Charleston, SC 29424, USA
pbuhler@cs.cofc.edu
[2] University of South Carolina, Computer Science and Engineering
Columbia, SC 29208, USA
vidal@sc.edu

**Abstract**. We describe a technique for providing agent software with dynamically configured capabilities. These capabilities, described with DAML-S, can represent atomic or orchestrated Web Services. The DAML-S specification will be transformed into an executable program written in a composition language named Piccola. When executed, the composite service will be available as a semantically described behavior within a FIPA compliant agent.

## 1 Introduction

Software development needs to progress from handcrafted, line-at-a-time techniques to methodologies that support reuse of existing software assets. In other words, software development needs to shift from paradigms that are purely creational to others that support compositional approaches. Traditional software engineering methodologies are giving way to new software development paradigms. Component-based software engineering and agent-oriented software engineering are two paradigms that are garnering attention. Although typically thought of as separate disciplines, it is likely that they are not only related, but also ultimately dependent upon one another. In the future, passive software components will be liberated by the proactive and social nature of agents. In effect agent-based technologies provide the mechanism for components to seek work, enter into cooperative agreements and thus otherwise address the requirements of dynamic, heterogeneous environments. From a multi-agent systems perspective, social agents that have access to an ontology-backed semantic description of the behaviors and services from which they are composed, should be better able to proactively coordinate themselves.

## 2 Emerging Approaches to Workflow Automation

A workflow is a process "during which documents, information or tasks are passed from one participant to another in a way that is governed by rules or procedures" [3]. From a workflow perspective, a composite software system can be viewed as a sequence of services operating upon data. Ideally these services should be language, platform and location independent. Such services would then be interoperable, where interoperability is characterized by the “ability of two or more software components to cooperate despite differences in language, interface, and execution platform” [13].

A new class of interoperable, web-enabled software services is emerging. These services are known as Web Services. “A Web Service is a software application identified by a URI, whose interfaces and binding are capable of being defined, described and discovered by XML artifacts and supports direct interactions with other software applications using XML based messages via Internet-based protocols” [12]

There are several ongoing initiatives that are defining compositional notations for Web Services. These notations express the flow of control and data across a collection of Web Services whose choreography performs a workflow. Recently IBM, Microsoft and BEA released BPEL4WS (Business Process Execution Language for Web Services). BPEL4WS is a specification for coordinating business process over the web [1]. In the future, the release of BPEL4WS may be viewed as a watershed event as it represents the first step toward market consolidation, as it replaces IBM’s WSFL and Microsoft’s XLANG specifications.

### 2.1 Agent-based Workflow Approaches

If a collection of sociable agents, representing individual services, cooperate and coordinate they would have the capability to enact any workflow that is composed of the represented services. In other words, agents have the capability to dynamically form social structures through which they share commitments to the common goal of workflow enactment. The individual agents, through their coordinated interactions achieve globally coherent behavior; they act as a collective entity known as a multi-agent system.

Workflow enactment by a multi-agent system is an example of cooperative problem solving. “Cooperative problem solving occurs when a group of autonomous agents choose to work together to achieve a common goal” [14]. For cooperative problem solving to occur, an agent in the multi-agent society must recognize that the best path to achieving a goal is to enlist the help of other agents. Social commitments arise when one agent makes a commitment to another. Typically a social commitment comes about due to a social dependency. As defined in [7, pg 113] a social dependence can be defined as:

$$(SocialDependence\ x\ y\ a\ p) \equiv (Goal\ x\ p) \wedge \neg(CanDo\ x\ a) \wedge (CanDo\ y\ a) \wedge ((DoneBy\ y\ a) \Rightarrow Eventually\ p)$$

*Meaning agent x depends on agent y with regard to act a for realizing state p, when p is a goal of x and x is unable to realize p while y is able to do so.*

As indicated, for such a social dependency to be established, agent $x$ and agent $y$ must be able to reason about their ability to perform act $a$, and have knowledge that the performance of $a$ will establish state $p$. The concept of first-order ability, as introduced in [14, pg 150], states that for agent $x$ to have first-order ability regarding the establishment of state $p$, it must know explicitly whether $\exists a$((CanDo $x$, $a$) $\wedge$

((DoneBy $x$ $a$) $\Rightarrow$ Eventually $p$)). If agent $x$ desires to achieve state $p$, but knows $\neg$(FirstOrderAbility $x$, $p$), then it must solicit assistance in order to attain the goal.

## 3 CONFLUENCE

It is the belief of the authors that the semantic web and the emergence of a Web Services component model can facilitate agent-based workflow management in open environments. If agents are used to wrap semantically described Web Services, then the semantic service descriptions become the basis for determining the agent's first-order abilities. Likewise, a common semantic markup for Web Services will facilitate effective communication between agents. We intend to build an experimental system that will utilize DAML-S and a composition language named Piccola.

### 3.1 DAML-S

The semantic web initiative is developing technologies for locating web resources based upon their semantic content. Included in this vision is DAML-S, a specification for providing semantic markup for Web Services. DAML-S is being designed to support the following Web Service related tasks: discovery, invocation, composition and interoperation, and execution monitoring [2]. DAML-S provides a machine-interpretable, ontology-backed semantic description of both atomic and composite web-services. As stated in [9], the markup provides:

> *declarative advertisements for service properties and capabilities which can be used for automatic service discovery;*
>
> *declarative APIs for individual Web Services that are necessary for automatic Web Service execution; and*
>
> *declarative specifications of the prerequisites and consequences of individual service use that are necessary for automatic service composition and interoperation*

DAML-S has the expressive power to encapsulate the composition of several services within a single service description. If an agent could enact a composite service as a behavior, it is intuitive that this will expand the agent's first-order abilities. Expanded first-order abilities will help the agent preserve its autonomy by reducing its social dependencies. As agents reduce their social dependencies, they create efficiencies across the operating environment. In effect, this approach is analogous to business process reengineering whose typical goal is to reduce transactional costs while providing the same or better service. Providing agents the capability to enact services described in DAML-S streamlines the workflow, thereby increasing the agent's goal-attaining efficiency by reducing the need for cooperative problem solving in multi-agent environments.

In DAML-S, a composite service can be recursively decomposed into a set of atomic services. Control constructs are provided by DAML-S to orchestrate the services that compose the workflow. Enactment of DAML-S described workflows is a difficult problem that has not been extensively studied; however, some initial work has been done. The DAML-S execution semantics presented in [5], were inspired by Milner's π–calculus. The π–calculus is useful for modeling systems of concurrent, communicating and mobile processes [11]. Milner's work has also been inspirational to the development of a composition language named Piccola.

### 3.2 Piccola

The development of specialized programming languages for expressing the composition of components is a recurring idea. Early evidence is provided by the utility attributed to UNIX shell scripting. The pipes and filters architecture of the UNIX shell in combination with a scripting language demonstrate the power of flexible composition via the pipelining of streams and commands.

In [10], the authors introduce the rationale and requirements for a general purpose composition language. The authors describe a composition language as providing the integration framework between the computational and compositional views of a system. The composition language requirements proposed by the authors are designed to support open systems development, where openness is characterized by the need for recomposability in the face of changing system requirements. The authors propose the development of a composition language using the π-Calculus as a theoretical foundation. This requirements groundwork ultimately results in the publication of [4, 8], which describe the composition language named Piccola. A platform neutral implementation of the Piccola language exists, it is Java-based and is named JPiccola.

## 4 Agentcities Related Research

The Agentcities Initiative intends to provide a platform to demonstrate the interoperation of independently authored agents that are geographically dispersed and executing within heterogeneous environments. The interoperation is accomplished through the use of open systems technologies and protocols. The protocols utilized in the Agentcities framework are those defined by the FIPA standards. It is the intent of the authors to use Agentcities as a research platform for the delivery of contextually appropriate Web Services, where the context is defined by geographic location.

The architecture for the proposed research is found in Figure 1. The architecture is designed for scalability, from mobile PDA devices with wireless connectivity to resource-rich server class systems. The architecture is designed to be compatible with existing and emerging open standards; as such interoperability within open agent societies and Web Services is maximized.

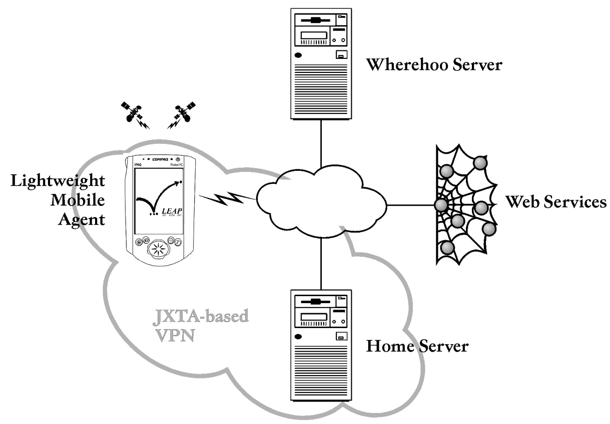


Fig 1. Proposed Architecture.

The major components of the architecture are:

- a Lightweight Mobile Agent implemented with LEAP [6]. The platform is an IPaq 3675 with a dual slot expansion pack; the expansion pack will hold an 802.11b wireless network card and a GPS receiver.
- the Wherehoo server [15] will store DAML-S descriptions of services associated with specific geographic locations. Wherehoo will return contextually appropriate DAML-S descriptions based upon the physical location of the mobile device.
- a Home Server will provide a Charleston, South Carolina, USA node to the Agentcities network, DAML-S to Piccola translation services, and a Piccola execution engine.
- the web of services will provide a dynamic set of behaviors for use by the mobile agent.

Operationally, the mobile agent will receive its absolute GPS position from the onboard GPS receiver. The location will be consumed by an internal behavior that will communicate with the Wherehoo server. The Wherehoo server will return a set of DAML-S descriptions for services that are appropriate within a physical region. The region is defined as a circle with selectable radius, whose center point is the current location. Each of the DAML-S descriptions will be passed to the Home Server where they will be transformed into Piccola programs. . It is anticipated that the transformation will leverage the Transformation API for XML (TrAX). A Piccola execution engine will execute the programs on the Home Server. The executing Piccola programs will communicate with the mobile agent via JXTA protocols and unidirectional pipes; JXTA provides the discovery services to allow the mobile agent to find the pipe endpoints on the Home Server. The result is the delivery of contextually appropriate Web Services to the mobile agent who views them as semantically described behaviors.

## 5 Acknowledgement

This work is supported by the U.S. National Science Foundation under grant IIS 0092593 (CAREER award).

## References

[1] XML Cover Pages. Business Process Execution Language for Web Services (BPEL4WS), http://xml.coverpages.org/bpel4ws.html.
[2] The DAML Services Coalition. DAML-S: A Semantic Markup For Web Services, http://www.daml.org/services/daml-s/2001/10/daml-s.pdf.
[3] Workflow Management Coalition. Introduction to the Workflow Management Coalition, http://www.wfmc.org/about.htm.
[4] Achermann, F., Lumpe, M., Schneider, J.-G. and Nierstrasz, O. Piccola - A Small Composition Language. in Bowman, H. and Derrick, J. eds. *Formal Methods for Distributed Processing: A Survey of Object-Oriented Approaches*, Cambridge University Press, New York, NY, 2001, 403-426.
[5] Ankolekar, A., Huch, F. and Sycara, K. Concurrent Execution Semantics for DAML-S with Subtypes. In *The First International Semantic Web Conference (ISWC)*, 2002.
[6] Bergenti, F. and Poggi, A. LEAP: A FIPA Platform for Handheld and Mobile Devices. In *Workshop on Agent Theories, Architectures, and Languages (ATAL-2001)*, 2001.
[7] Huhns, M.N. and Stephens, L.M. Multiagent Systems and Societies of Agents. in Weiss, G. ed. *Multiagent Systems: A Modern Approach to Distributed Artifical Intelligence*, MIT Press, Cambridge, MA, 1999, 79-120.
[8] Lumpe, M., Achermann, F. and Nierstrasz, O. A Formal Language for Composition. in Leavens, G. and Sitaraman, M. eds. *Foundations of Component-Based Systems*, Cambridge University Press, New York, 2000.
[9] McIlraith, S.A., Son, T.C. and Zeng, H. Mobilizing the Semantic Web with DAML-Enabled Web Services. In *Semantic Web Workshop*, 2001.
[10] Nierstrasz, O. and Meijler, T. Requirements for a Composition Language. in Ciancarini, P., Nierstrasz, O.M. and Yonezawa, A. eds. *Proceedings of the ECOOP '94 Workshop on Models and Languages for Coordination of Parallelism and Distribution, LNCS 924*, Springer, New York, 1995, 147-161.
[11] Sangiorgi, D. and Walker, D. *The pi -calculus : a theory of mobile processes*. Cambridge University Press, Cambridge, England ; New York, 2001.
[12] W3C Web Services Architecture Working Group. Web Services Architecture Requirements, Working Draft 29 April 2002, http://www.w3.org/TR/2002/WD-wsa-reqs-20020429.
[13] Wegner, P. Interoperability. *ACM Computing Surveys*, *28*(1):285-287, 1996.
[14] Wooldridge, M.J. *Reasoning about rational agents*. MIT Press, Cambridge, Mass., 2000.
[15] Jim Youll. Wherehoo Technical Overview, http://www.wherehoo.org/about/wherehoo_technical.html.