Location Dependent Service Delivery to Resource Limited Mobile Devices

Paul A. Buhler
College of Charleston
Dept. of Computer Science
66 George Street
Charleston, SC 29424, USA
pbuhler@cs.cofc.edu

José M.Vidal
University of South Carolina
Computer Science and Engineering
Columbia, SC 29208, USA
vidal@sc.edu

ABSTRACT

This position paper describes a novel architecture that will imbue agent software with dynamically configured capabilities. These capabilities, described with DAML-S, can represent atomic or orchestrated 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 the FIPA compliant agent. The proposed architecture is designed for scalability, from mobile PDA devices with wireless connectivity to resource-rich server class systems.

1. INTRODUCTION

It is the belief of the authors that the advent of 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 [13, pg 150]. Likewise, a common semantic markup for Web Services will facilitate effective communication between agents. DAML-S and Piccola are two emergent technologies that will be leveraged to deliver Web Services to FIPA compliant agents.

2. DAML-S

Composing dynamic, distributed systems presents many challenges. One of the greatest challenges is having accurate, just-in-time, information about the existence, availability, composition model and constraints that govern the use of a component [8, 11]. 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 [1]. DAML-S provides a machine-interpretable, ontology-backed semantic description of Web Services.

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 help the agent preserve its autonomy when compared to multi-agent enactment strategies. The wrapping of a DAML-S described composite service empowers an agent with greater capability, while at the same time creating efficiencies across the operating environment. One of the goals of business process reengineering is to reduce transactional costs while providing the same or better service. Since the agent has expanded first-order abilities, it can reduce its social dependencies along with the overhead associated with cooperative problem solving.

3. 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.

Neirstrasz, et al. introduce the rationale and requirements for a general purpose composition language in [10]. The authors view is that a composition language provides the integration framework between the computational and compositional views of a system. The resultant composition language, named Piccola, utilizes Milner's π-Calculus as its theoretical foundation [3, 4, 9]. JPiccola provides a Java-based, platform neutral implementation of the Piccola language.

4. RESEARCH PLAN

Software architecture research is given to the understanding of the structure of software systems, especially the relations among subsystems and components [12]. Components are inseparable from architecture in that their use is typically dependent upon infrastructure services that constrain the compositional relationships they can support [7]. The proposed architecture integrates thoughts from both component-based software engineering and agent-oriented software engineering.

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. 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 [14] will store a DAML-S description of services associated with a particular geographic location. Wherehoo will return contextually appropriate DAML-S description based upon the physical location of the mobile device. The Wherehoo server will provide service discovery services, allowing the focus of the research to remain on the DAML-S transformation process.
- 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 to the mobile agent.

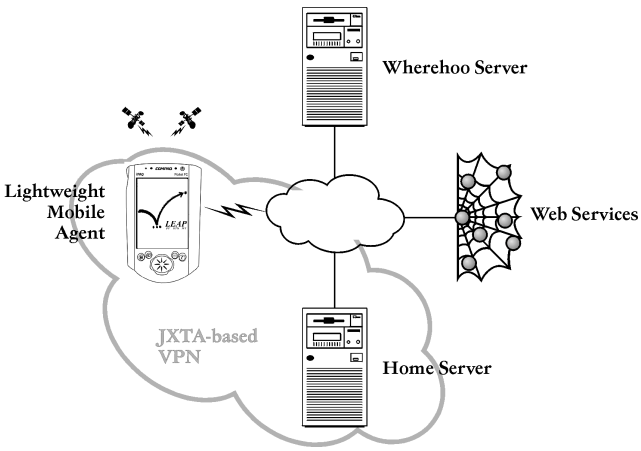

Figure 1. Proposed Architecture

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 spatially appropriate. 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) [2]. A Piccola execution engine will run the programs on the Home Server. The executing 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 accesses them as semantically described behaviors.

Alternatively, the Piccola processes could be wrapped in an agent and registered with the Home Server's Agent Management System (AMS). The Home Server's Directory Facilitator (DF) could then be used to link the mobile agent with its agent-based behavior. However, when the behavior is no longer required, the mobile agent cannot teardown the remote agent without violating its autonomy. Although the mobile agent and Piccola processes communicate as peers, a master-slave relationship exists between them. When the mobile agent no longer requires a behavior, it will send a teardown request to the Piccola process, which will end execution.

4.1 Limitations

The centerpiece of the research is the DAML-S to Piccola transformation and subsequent execution of the Piccola program. A robust translation service would prove useful in numerous domains; however, this work is focused on a subset of DAML-S known as DAML-S Core, as defined in [5]. It should also be noted that the proposed architecture leverages the Wherehoo Server for DAML-S discovery. Although this mechanism is suitable for the described system, its usefulness will be limited in other domains.

5. REFERENCES

[1] The DAML Services Coalition. DAML-S: A Semantic Markup For Web Services, http://www.daml.org/services/daml-s/2001/10/daml-s.pdf.

[2] The Apache XML Project. Transformation API for XML, http://xml.apache.org/xalan-j/trax.html.

[3] 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.

[4] Achermann, F. and Nierstrasz, O. Application = Components + Scripts - A tour of Piccola. in Aksit, M. ed. *Software Architectures and Component Technology*, Kluwer Academic Press, 2000.

[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] Brown, A.W. and Wallnau, K.C. The Current State of CBSE. *IEEE Software*, *15*(5):37-46, 1998.

[8] Garland, D. Software Architecture: a Roadmap. In *The Future of Software Engineering*, ACM Press, 91-101, 2000.

[9] 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.

[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] Nierstrasz, O., Schneider, J.-G. and Achermann, F. Agents Everywhere, All the Time. In *Workshop on Component-Oriented Programming at ECOOP2000*, 2000.

[12] Shaw, M. The Coming-of-Age of Software Architecture Research. In *Proceedings of the 23rd International Conference on Software Engineering*, IEEE Computer Society, 657-664a, 2001.

[13] Wooldridge, M.J. *Reasoning about rational agents*. MIT Press, Cambridge, Mass., 2000.

[14] Jim Youll. Wherehoo Technical Overview, http://www.wherehoo.org/about/wherehoo_technical.html.