Abstract

The current state of the art in agent technology sees that several implementations of agent frameworks exist. However, there is little agreement on the terms and concepts used to describe such systems, which is a significant barrier towards adoption of these technologies by industry, military and commercial entities. A clear definition of terms and concepts at an appropriate level of abstraction is needed to facilitate discussion, evaluation and adoption of these emerging agent technologies. In this paper, we argue that a reference model for agent-based systems can fill this need. We discuss what a reference model is and why one is needed for agent-based systems. While the complete model is a work in progress, we present a preliminary version to motivate further discussion from the agents community at large. It is our hope that ultimately a wider community of practice will assume responsibility for the standardization similar to the way that the well-known seven-layer Open Systems Interconnection (OSI) reference model was a driving force underlying communications standards.

1 Introduction

The ultimate goal of the Agent-Based Systems Reference Model (ABSRM) is to provide a technical recommendation for a reference model for those who develop and deploy systems based on agent technology. The ABSRM should allow for existing and future agent frameworks to be compared and contrasted as well as to provide a basis for identifying areas that require standardization within the agents community. As such, the aim of the ABSRM is to

- establish a taxonomy of terms, concepts and definitions needed to compare agent-based systems;
- identify functional elements that are common in agent-based systems;
- capture data flow and dependencies among the functional elements in agent-based systems;
- specify assumptions and requirements regarding the dependencies among these elements.

As a reference model, the ABSRM will make no prescriptive recommendations about how to best implement an agent-based system; nor is the objective to advocate for any particular approach, architecture or framework. In its broadest sense, an agent-based system for the purposes of the ABSRM simply describes a software platform for building agents and supporting their communications and collaboration. An agent-based system may consist of many different kinds of agents operating across a heterogeneous set of platforms and hosts.

One novel aspect of our approach is to create the reference model based on a forensic analysis of existing agent-based systems. The reference model developed in this document is based on static and dynamic software analysis of existing agent frameworks. Examined frameworks include Cougaar, JADE, RETSINA, and others. Anyone building an agent framework would have to recreate or reproduce some portion of the functionality or components in these existing frameworks (i.e., to enable communications, to enable agent startup and shutdown, etc). By analyzing existing frameworks and the agent-based systems they can be used to build, we can avoid the debate concerning “what is an agent” and simply document the existing state-of-the-art systems that are called agent-based. The model aims to document a superset of the features and functional concepts in the set of existing agent frameworks. Given that there is significant variation between existing frameworks and the functions they may provide, the reference model should describe at an abstract layer the complete set of functional components across all examined agent frameworks.

It is important to note however that the reference model is not confined to being a description of capabilities of existing systems—it serves as a basis for situating the complete set of functions that anyone may want or need to have in an agent-based system. For example, security for mobile agent code is currently a vastly challenging problem that lacks a satisfactory solution. However, the lack of any established, uniform and generally accepted security system for mobile
agents does not prevent the reference model for including a description of the security functions and facilities that an agent-based system may provide.

2 What is a Reference Model?

A reference model provides appropriate abstractions for facilitating adoption, adaptation and integration of evolving technologies [6]. Any generic model that has specific examples can be considered to be a reference model. Reference models are known to play a key role in understanding a given domain, establishing the domain as a scientific discipline, facilitating collaboration and promoting competition towards maturing technology relevant to the domain. Reference models emerged since the 1980s as a result of the success of the Open Systems Interconnection seven-layer reference model (ISO/IEC 7498-1:1994: Open Systems Interconnection Basic Reference Model) [3] [4] [5] that revolutionized the way communications systems developed. This model was developed as an International Standards Organization (ISO) effort. Another successful example of a reference model that has been developed by the ISO is one for archiving systems and is known as the Reference Model for an Open Archival Information Systems (OAIS) (ISO 14721:2003). In motivating and developing a reference model for agent systems, we draw heavily on these examples of existing successful reference models.

As these existing reference models demonstrate, reference models do not prescribe how functions and systems should be implemented. Instead, reference models provide the patterns of the solution for transforming vague notions into real-world implementation. Reference models simplify problem solving, to enable others to practice their discipline with a solid foundation. Software professionals, in particular use reference models to better understand abstractions and their potential for reuse.

3 Why a Reference Model?

Reference models are a necessity in a confusing, rapidly changing technology environment. As noted in [6], reference models are becoming more commonplace in fields of various human endeavors. The power of compelling reference models of knowledge can not be underestimated as a tool for technical leadership facilitating conclusions from a sound understanding of the problem space and solution domain. Thus, we argue that a reference model would be very useful tool to identify, assess and facilitate R&D and acquisition of agent technology for a wide variety of applications.

One particular application of agent-based systems is in military domains. Agent-based systems are being proposed to enhance and automate applications for collecting, presenting, storing, producing and sharing domain information. A future force is envisioned to be highly autonomous, modular, scalable, and flexible through the agentization of applications and services. This is especially true for complex system of systems (SoS). SoSs are large-scale, net-centric and include a variable mix of multi-department heterogeneous intelligent agents, humans-in-the-loop, and unmanned autonomous components and subsystems. Examples of complex SoSs in the US Army are the Army Battle Command System of the Current Force [1] and the Future Combat System of the Future Force [2]. Systems that would be found in a Joint Service (Army, Navy, Marine and Air
Heterogeneous intelligent agents promise to enable conflict resolution between and among applications and services engaged in competition, negotiation, mediation and arbitration, to assist humans-in-the-loop, and to control unmanned autonomous systems and robots. Agents are anticipated to play an important role in realizing dynamically varying mixed initiative capabilities to command and control manned as well as unmanned assets. As systems become increasingly complex, modularity as promoted by agent technology will become the key to reuse, scalability, and an open architecture. In addition, these design goals are a key to a manageable and affordable transformation from current to future force capabilities.

A reference model for intelligent agents would motivate the benefits of the technology by formalizing key concepts of both behavior and structure essential to enable agent technologies to reduce information collection, storage and sharing latency, workload, presentation overload and clutter for Battle Commanders and their staff. The recognition of the need and the investment to develop a unifying ontology for intelligent agent software across the domains of the systems in an enterprise-wide System of Systems (SoS) are shown to be crucial if not pivotal to the success of such SoS engineering efforts which are inherently multi-disciplinary and collaborative.

4 Towards A Reference Model

We describe a preliminary, partial reference model for agent-based systems. The portion of the model we present here is a high-level view organized as a set of layers. A more complete description of the current reference model is given in [7].

An Agent-Based System is comprised of agents and their supporting framework and infrastructure which provide fundamental services and operating context to the agents. Our model defines framework, platform and host layers, which mediate between the agents and the external environment. This layered model can be organized vertically as shown in Figure 2. Each layer is described as follows:

- The Agents layer consists of agents that perform computation, share knowledge, interact and generally execute behaviors in order to achieve application level functionality. We make few assumptions about the Agent layer except to state that agents are situated computational processes—instantiated programs that sense and effect an environment in which they exist. We make no assumptions about the internal processing structures of an agent. An agent could be built with a complex cognitive model or it could be a simple rule-based system. Given the vast array of tasks envisioned for agent systems, it is not the role of a reference model to limit or define what an agent is.

- The Framework layer provides standardized functionality specific to supporting agents. A framework typically provides support services such as conflict management, directory and naming services, security, and agent administration services such as monitoring and allocating resources to the executing agents. Figure 3 shows some examples. The major benefit of employing an agent framework is to provide standardization of services and functionality to agents that exist within the framework. The end result is that agents written within a particular framework are easily interoperable with one another. In other agent-based systems, the framework may be trivial or merely conceptual, for example if the services are merely a collection of system calls or are compiled into the agents themselves.

- The Platform layer provides more generic computing infrastructure. The platform contains the software components that are available to the agent framework, but are not packaged along with it. Some examples are shown in Figure 3. Elements such as operating systems, user interface libraries, database software, device drivers, and message transport or socket libraries are in this layer. These services are often provided by third parties and it is unlikely that an agent-based system will provide its own implementation of the platform functions.

- The Host layer contains the hardware devices on which the above layers operate. This layer includes not only the physical computing devices such as a desktop computer or hand-held device, but also the hardware that provides interaction with the environment such as robot sensors and effectors, cameras, displays, GPS receivers, etc. Some examples are shown in Figure 3

- The Physical world layer encapsulates the physical environment in which the agent-based system exists and operates.

Each layer can support many entities from the layers above it—many agents may execute on a single framework, many frameworks may execute on a single platform, and so on.

Figure 4 depicts an alternative view of an agent-based system. The figure organizes the interactions and communications between entities (agents, frameworks, hosts) at varying levels of abstraction. At the lowest level (physical network), hosts can communicate over a physical transport medium which can be either wired or wireless. A wireless communication medium is shown in our figure as an example. Going one level higher, connectivity between hosts
is depicted at the network layer. Connectivity at this layer is determined by routing, naming/addressing or other network services. At the third higher level, we show the connectivity between agent framework instantiations on different hosts. Each host is now represented by a larger dashed oval and each framework instantiation is represented by a smaller solid oval. By showing framework ovals inside a host oval, we depict multiple frameworks running on a given host. Lines connecting the frameworks instantiations show that they can communicate to share information and services. Finally, at the highest level we show the communications that occur between framework instantiations of the same type (e.g., Cougaaar, JADE, etc). In this way, we can depict that agents with common application goals can form societies that communicate in order to provide specific application functionality.

5 On-going Work

Our goal is to continue refinement of the reference model in consultation with the broader agents community. We will be making available an on-line interactive discussion forum, similar to a Wiki, for facilitating further development of the ABSRM.

Acknowledgments

We thank Christopher Dugan, Moshe Kam, Joseph Kopena, Robert Lass, Spiros Mancoridis, William Mongan, Jeff Salvage, Evan Sultanik, for significant contributions to this work. We also thank Tedd Gimber, Bernard Goren, Michael Huhns, James Odell, Randy Reitmeyer and Todd Urness for useful discussions.

References