

Supporting Internet-Scale Multi-Agent Systems

Niek Wijngaards Benno Overeinder Maarten van Steen
 Frances Brazier

Dept. of Computer Science, Faculty of Sciences, Vrije Universiteit
Amsterdam, de Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
{niek,bjo,steen,frances}@cs.vu.nl

Although current research on agent systems focuses on relatively small-scale agent systems, soon vast numbers of agents will be deployed in large-scale agent systems, e.g. on the Internet. Large-scale agent systems need to be extensible, heterogenous, interoperable, and scalable. Scalability entails systems to scale (in terms of the number of agents and available resources) almost immediately without noticeable loss of performance, or considerable increase in administrative complexity [1].

In Data and Knowledge Engineering, 2002, 41(2-3): 229-245.

AgentScape

The overall design philosophy behind AgentScape is “less is more” and “one size does not fit all.” The AgentScape middleware provides minimal but sufficient support for agent applications. In addition, the middleware is adaptive or reconfigurable such that it can be tailored to specific applications or operating systems/hardware platforms.

Agents and objects are basic entities in AgentScape. Agents are active entities in AgentScape that communicate with each other by message-passing. Objects are passive entities that contain information and can be manipulated. AgentScape objects are Globe objects [3] which have their own replication strategies to distribute their internal state.

Basic services in AgentScape are naming and location services for agents, objects, and locations. These services enable agents to find and contact other agents or objects in the distributed multi-agent system, and to migrate to other locations.

Fig. 1 presents a model of AgentScape from the agent perspective, that is, the location comprising the middleware and the resources are represented by a location manager agent and resource objects. Calls from an agent to the middleware are modeled by requests to the location manager agent to, for example, create an agent or move an agent. Information about resources residing at the location can be retrieved by binding to the resource objects, which are *local* distributed objects. These objects can be accessed only within the location they reside, not from outside the location.

For development of agent applications, an application programming interface (API) and a runtime system (RTS) are provided, see Fig. 1. The default API and RTS can be extended to provide a higher-level application programming interface with, for example, a model that offers more structure and semantics to the agent application developer.

Within AgentScape, management of large-scale agent systems is an important issue, including not only life-cycle management of agents, but also management of security

and authentication, middleware configuration, and resources. As centralized management mechanisms are not applicable (scalability), other approaches need to be considered.

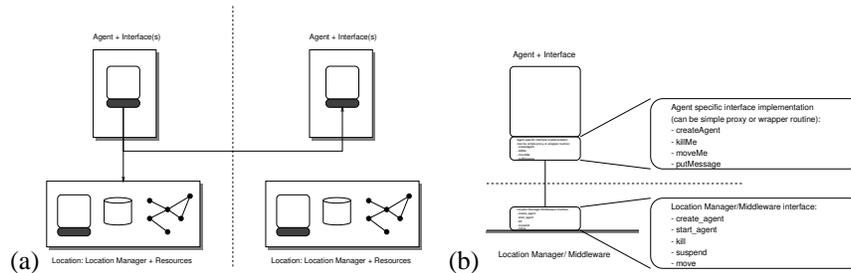


Figure 1: (a) The AgentScope model from an agent's perspective. (b) The relation between agent-specific interfaces vs the aos interface.

Discussion

AgentScope addresses some important research issues regarding the realization of large-scale agent systems, such as scalability, language/platform independence, and interoperability. The AgentScope model effectively supports both Artificial Intelligence and Computer Systems research interests. From the AI point of view, AgentScope is an extensible framework for the development of large-scale agent systems, such as more structured models for agents' environments. From the CS point of view, AgentScope provides an experimental framework for studying of new approaches in large-scale agent systems, such as scalable directory services, security, and interaction models.

Supporting large-scale agent systems involves solving numerous problems on both the agent-level and the system-level [2]. In this research area, much progress may be achieved by cooperation between researchers from both the AI and CS communities.

Acknowledgments

This work is supported by the NLnet Foundation, <http://www.nlnet.nl/>. The authors wish to acknowledge the contributions made by Andy Tanenbaum and Guido van 't Noordende.

References

- [1] B. Neuman. Scale in distributed systems. In T. Casavant and M. Singhal, editors, *Readings in Distributed Computing Systems*, pages 463–489. IEEE Computer Society Press, 1994.
- [2] A. S. Tanenbaum and M. van Steen. *Distributed Systems: Principles and Paradigms*. Prentice Hall, Upper Saddle River, NJ, 2002.
- [3] M. van Steen, P. Homburg, and A. S. Tanenbaum. Globe: A wide-area distributed system. *IEEE Concurrency*, 7(1):70–78, January–March 1999.