Where Are All the Intelligent Agents?

Editor: James Hendler
Rennselaer Polytechnic Institute
hendler@cs.rpi.edu

Intelligent Readers,

In the late 1990s, many of us believed that the large-scale deployment of "intelligent agent"-based computing was right around the corner. In the early 2000s, much US and international research funding focused on making this happen. This magazine and many others had hugely popular special issues on agents, and academic conferences on agent-based computing were abuzz, with many of us "old timers" beginning to believe that the time had really come. But now, looking at what's hot on the Web, in IT development, and in venture capital circles, I find myself shaking my head and wondering, "Where are all the agents?"

Working toward interoperability

I was personally very involved in this agent buzz. Starting in late 1998, I took a few years out of my usual academic life to work at Darpa. I went there to help develop and enhance the funding profile in agent-based systems. I had been working in the intelligent-agents community for well over a decade, and Darpa felt it would be useful to bring in an academic researcher to help move work in the area forward. During my time there, I helped oversee tens of millions of dollars of government funding for research in this area, making me, for a time, the largest funder of agent systems research in the US, and probably the world.

When I started at Darpa, I took over the Control of Agent-Based Systems program, which Tom Garvey and Doug Dyer, among others, had started. CoABS focused on creating software infrastructure to support the large-scale deployment of agent-based programs. Over the next couple of years, the program refocused on developing middleware for agent-based systems—essentially an interoperability program aimed at making different agent architectures work together.

The reason for this redirection was that researchers at that time were exploring a number of different agent frameworks and several relatively successful architectures had been developed. However, these systems didn't play well together, and it was clear that for agent systems to be successful in the real world, we would need to assume more heterogeneity. We needed to create an infrastructure that would let these different agent systems find and register callable services. So, you would be able to use the individual agent architectures to wrap existing systems for use as agents, to provide the "autonomy" needed for specific applications and to essentially provide services. When a system didn't have the needed capabilities, it could find and link to services offered by other agent systems.

Our response was to create the CoABS Grid middleware package. The CoABS Grid let agent applications communicate with each other and facilitated the wrapping of legacy systems so that they could communicate like agents (that is, provide services for agent-based systems also using the Grid). The Grid code was based on the Jini network architecture, which at the time was clearly the best choice for such middleware development. The CoABS Grid has validated many ideas behind current service-oriented middleware architectures. To date, it has been used in both military and commercial systems. Software developed from it is available from Global Infotek, under the name Intelligent Services

We'd Like to Hear from You

Letters to the Editor:
Send letters, including a reference to the article in question, to intelligent@computer.org. Letters will be edited for clarity and length.

Articles:
If you're interested in submitting an article for publication, see our author guidelines at www.computer.org/intelligent/author.htm.

1541-1672/07/$25.00 © 2007 IEEE
Published by the IEEE Computer Society

Editorial Board Transition

Robert Laddaga has stepped down as associate editor in chief. He will continue to serve on our advisory board. We thank him for his years of dedicated service.

Correction

The reference list in the Mar./Apr. theme feature article, "Autonomy and Common Ground in Human-Robot Interaction: A Field Study," by Kristen Stubbs, Pamela J. Hinds, and David Wettergreen (pp. 42–50), omitted reference 7. The reference is S.R. Fussell and R.M. Krauss, "Coordination of Knowledge in Communication: Effects of Speakers' Assumptions about What Others Know," *J. Personality and Social Psychology*, vol. 61, 1992, pp. 378–391.

Layer, or ISL (www.globalinfotek.com/what_is_isl.shtml).

Perhaps more important, many ideas behind the CoABS Grid have become popular in the Web services framework. The idea of wrapping software to provide services that other systems can invoke remotely is this approach's core, and industry has adopted the middleware approaches pioneered in CoABS and other research as a key software development concept. Instead of needing proprietary technology to wrap legacy systems, we've got SOAP, WSDL, and a stack of service languages that have become standards. The registry of services, the "choreography" of multiple service providers, and other ideas that were coming out of the research community in the late 1990s are now mainstream. In short, mainstream software approaches can now provide the agent community's interoperability infrastructure needs.

The knowledge-engineering bottleneck

While at DARPA, I realized a second problem with existing agent systems: even with the infrastructure, interoperability at the data level was still difficult. The development of agent-based systems was clearly being constricted by the same knowledge-engineering bottleneck that has been the choke point for the widespread application of expert and intelligent systems. Trying to overcome this led to the creation of the DARPA Agent Markup Language (DAML) program. The basic concept was that if we could make knowledge shareable and easily distributed (linkable came later), we could make it so that agents could share not only services but also vocabularies.

This wasn't a totally new idea at DARPA. Earlier programs had tried to develop standards for knowledge interchange, and the Knowledge Query Manipulation Language (KQML) laid some of the groundwork for such a language's needs. DAML's primary innovations were to base these earlier ideas in new Web languages so as to enhance ease of deployment, simplify some of the representational commitments, and provide some common tools, to make ontology development easier. The DAML program also sought to develop a de facto standard for these Web ontologies, and EU and US government funding agencies jointly created a committee to develop a suitable language. This language, DAML+OIL, was published in December 2000. In a couple of years, it had created enough interest that industrial, academic, and government participants agreed to standardize it under the auspices of the World Wide Web Consortium. The resulting Web Ontology Language, OWL, became a standard in February 2004 and has become a key language for the Semantic Web.

Scant evidence of deployment

So here's what I realized. The key obstacles to the wider deployment of agent-based systems were identified early on as the needs for interoperability and intercommunication.

We now have Web service standards, supported by the largest software development and support companies, which provide for many of the interoperability needs we identified. The Semantic Web is also seeing wide deployment and support from some of the larger data-providing companies. Open source toolkits and tens of thousands of OWL ontologies are available to ease domain engineering. Many large Web providers are making access to their systems available through some sort of service interface or in easily programmable ways. Technologies transitioning from research to industry also include data access for Semantic Web resources, rule-based Web languages, and even expressive logics for the high-end knowledge representation needs of some applications.

What we don't seem to have are intelligent-agent-based systems in any serious way!

While there's clearly still an active research community in agents, I see no evidence for the imminent widespread use of this technology such as we were promising a decade ago. The main agent conference, AAMAS (the International Joint Conference on Autonomous Agents and Multiagent Systems), seems to be doing fine in terms of attendance, and the applications track has a fair amount of interesting work, but few of the published papers talk about any sort of real-world deployment. The Web site that was serving as the agent community's primary portal, the University of Maryland, Baltimore County's AgentWeb, seems to have last been updated sometime in 2005. The bulk of the papers I can find published since then are filled with all kinds of wonderful theory but not much on deployed applications. In fact, looking back over the special issues on agents that we've been publishing in this magazine every couple of years, I see few if any papers that report on systems that have been fielded "in the wild," large scale or not.

So what happened? When I began my work on the Semantic Web, agents were a key motivation. In fact, if you look back at the widely cited article "The Semantic Web," which Tim Berners-Lee, Ora Lassila, and I wrote for *Scientific American* when we started working together (May 2001—it's been a while!), you'll see agents were clearly a theme. The rest of the ideas in that article are now seeing widespread deployment, but I ask again:

Where are all the agents?

James Hendler