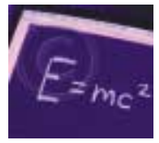
#62-1

your agent is calling!



In an agent-filled future, you will trust your agent to tackle a range of daily problems and tasks.
What these agents "look" like, how they interact with each other and us and what defines and limits their behavior are all questions HP Labs' Martin Griss is working to answer.

photographs by
David Toerge

#62-2

"The most recent agents I and several colleagues have built are focused on the personal space. The vision we have been pursuing is that of a "personal assistant" to a mobile person engaged in various kinds of personal or work activities."

For Martin Griss, Ph.D. in physics, Laboratory Scientist, and HP Labs' "Reuse Rabbi," the world of agents is less about holographic "butlers" that greet a Jetson-like family to report on the day's events and more about software components that can handle an assortment of tasks from ordering take-out to scheduling meetings.

The agents Griss works with may or may not have visible personas, but they will effortlessly shift roles and duties, adapt to current needs, make decisions based on context and allow users to potentially pack even more into an average day. Griss is particularly interested in how agents can and will work together, interacting with networks of agent systems as they perform their tasks.

When we talk about "agents," many of us imagine a "butler" type of presence that appears on screen, interacts with us and performs basic tasks to simplify processes. How "embodied" does the agent become?
In our work, we have largely focused on the functionality, customizability and robustness of the personal assistant and supporting agents. We have done some work on providing a more natural interface using speech and natural language — for example, we can say, "Please arrange a meeting with Reed and Craig next Friday," and voice output, especially to voice mail, but the bulk of our work has been using simple Web interfaces to create "one-button meetings," instead of using three or four separate Web interfaces and e-mail exchange with consequent error-prone information exchange.

We have not explored the role of an anthropomorphic persona or avatar. There are other research groups who have concentrated most of their effort on natural interfaces, gestures and animation to make the agent feel more flexible and more natural. It is, in fact, an interesting question as to whether these more natural interfaces increase our trust and comfort in using agents and in delegating increasing responsibility to them.

#62-3

What kinds of agents have you experimented with for personal use?
The most recent agents I and several colleagues have built are focused on the personal space. The vision we have been pursuing is that of a "personal assistant" to a mobile person engaged in various kinds of personal or work activities.

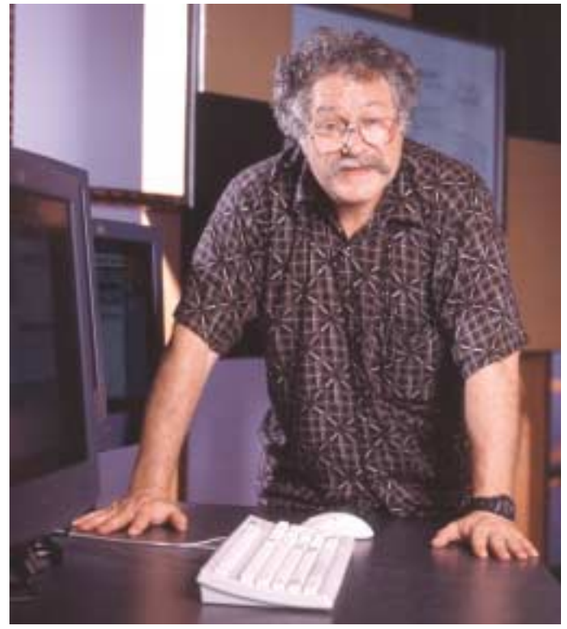
I am interested in tools that help me in dealing with the complexity of my daily life — both personal and professional. Like so many other folks, I am always on the go. For example, I might be at work when my personal assistant agent (PA) reminds me that it's time to meet my daughter to work on a photography project in the park and that I promised to bring lunch.

As I leave the office, I ask my PA to reschedule some meetings for the afternoon, send the document I was working on to my team and let my daughter know I will be a bit late since I have to pick up lunch. The assistant knows where I am meeting my daughter, knows what she and I like to eat and suggests a Chinese meal that can be prepared and ready for pickup as I swing by on my way to meet my daughter. I dialogue with the PA to adjust the menu. The PA orders and pays for the meal and summarizes some of my e-mail as it also gives directions, traffic conditions and has someone from the restaurant meet me at the restaurant loading bay as I drive by.

What are you working on currently?
Focusing on the personal and mobile aspects of agents, and for the moment downplaying the shopping and e-commerce aspects, we have been building a system we call "CoolAgent," which involves a group of personal, team and service agents and assistants. These agents and assistants work together to manage calendars, meeting rooms and teleconference lines to set up distributed meetings.

While our demo is targeted at the mobile professional and the workplace, the same kind of personal assistant with supporting agents could help set up meetings and events between myself and my friends, my family, PTA, to better achieve work/life balance as my several roles interact.

Each personal assistant has access to a rich profile of personal information about myself, knowledge on how and when to access my calendar, what my preferences are for certain kinds of meetings and how I like to be notified (by e-mail, pager, voice mail, etc.) in different circumstances. In order to schedule meetings on my behalf, or respond to a meeting request from some other assistant, my assistant may be authorized to look into my calendar, to respond with my availability for certain meetings and to reserve and confirm those meetings. At the same time, my assistant, and those of other participants, will negotiate with a meeting assistant, and perhaps a team assistant, to establish the best time, access the HP Labs' room reservation system to find an appropriate sized room with the right attributes and equipment and also to reserve a teleconference line if some of the participants will be attending remotely.



#62-4

In one of your papers, you define software agents as "personalized, continuously running and semi-autonomous, driven by a set of beliefs, desires and intentions (BDI)." What kind of BDI parameters might underwrite the behavior and activities of a user's agent?
There are a number of models of agent-oriented programming. In general, when a set of components are very loosely coupled, communicate by exchanging high-level structured messages (such as KQML, or FIPA, or ACL based on speech act theory) and exhibit significant autonomy (having goals, pursuing them independently and being able to find and request assistance and services from other such components), we call them agents.

It's convenient to use an anthropomorphic abstraction when decomposing a system into agents and when assigning responsibilities and programming an agent. We tend to ask questions like "What does the agent know? What does it want to do next? Can an agent know what another agent believes?" In some cases, this has been formalized into the BDI model, which can be used to reason about what other agents know and might do in

"In order to schedule meetings on my behalf, or respond to a meeting request from some other assistant, my assistant may be authorized to look into my calendar, to respond with my availability for certain meetings and to reserve and confirm those meetings."

certain situations; in other cases, the use is more informal. In our work, we are much more informal since we do not believe we can have total control over what agents written by different programmers will do.

So instead of thinking about BDI in regard to other agents, we use knowledge, goals and rules to program our agents and have them exhibit specific behaviors and follow prescribed patterns of message exchange (message protocols), which involve specific use of words (the ontology).

"We want our agents to be flexible and autonomous. That means they must be able to handle complex alternatives, ideally be able to handle situations that have not been fully detailed by the programmer and be able to adjust to changing situations and priorities."

How are BDI and artificial intelligence related?
First of all, we want our agents to be flexible and autonomous. That means they must be able to handle complex alternatives, ideally be able to handle situations that have not been fully detailed by the programmer and be able to adjust to changing situations and priorities. So while all of this capability can be programmed directly in a procedural language such as Java, it is much more convenient to use toolkits that allow us to program at much higher levels of abstraction.

These include a number of AI techniques: rule-based and knowledge-based software (using a rule interpreter such as Prolog, CLIPS or Jess), planners and goal-directed techniques (so that a goal can be stated at a higher level and decomposed into achievable sub-goals) and fuzzy logic and machine learning techniques (such as neural nets and Bayesian reasoning).

Not all agents need to use all or any of these techniques, but they make it easier to give the "illusion" of autonomy and intelligence. The challenge in using such techniques is making it easy for the user to understand what will happen under different situations.

Agent construction seems particularly tricky given standardization and interoperability issues. What steps need to be taken industry-wide to support a widespread system of interoperability that can support the construction of a robust network of software agents?
Even in the absence of agents, we have the ability to create new standards, new rules, new business models, new roles and common jargon and language for communicating. However, many of these have been represented informally, or at least in an ad hoc machine-inaccessible form (e.g., consider the tax code or legal system; EDI is more formal but 1-to-1 ad hoc). If we want autonomous intelligent agents and Web services to be able to dynamically discover each other, and establish communication and interaction, and then begin to negotiate, the standards and rules have to be represented in a more formal structure.

This is where work on ontologies, agent standards (like FIPA) and Web service standards (XML, SOAP, UDDI, ebXML, RosettaNet, semantic web) will play a significant part. These communication and representation standards are now beginning to emerge. Some parts will become ubiquitous, while others will be associated with particular e-systems or ecosystems; perhaps this will be an opportunity for translator agents and services.

The variety of agent systems that exist — Aglets, Grasshopper, Jackal, Voyager, Zeus, etc. — hints at an already robust agent-oriented landscape, but can all these different agents really communicate with one another and work together on a variety of planes?
Indeed this is a problem — I have found over 100 different agent systems, many with their own agent-modeling communication languages. However, recently FIPA has established a set of standards, which many of the newer agent systems try to follow. In particular, there is a standard for an agent communication language (ACL) and the model of a standard agent platform and agent services, which is intended to make it easier for separate agent systems to communicate. Also, recently a number of agent systems (such as Jade, Zeus, FIPA-OS, Grasshopper) have started to release Java-based open source, or at least "source-available," platforms.

These kinds of standards are increasing the possibility and probability of interoperation.

Finally, a large-scale, multinational interoperation effort, called AgentCities, is expected to act as a forcing function for the development and testing of ontologies, agent communication languages, agent platform and agent services. HP Labs is playing a key role in helping define AgentCities.

How far are we from having an agent in every home?
It depends on what we want the agent to do, and whether it comes bundled with a mobile phone, lives as a service in our ISP, telco or cable provider or is set up by the individual. Perhaps the agent is not "in every home" — instead, it lives in a secure place on the Web and delivers services to us through the Web or phone network. Better to say "when every person has their own personal assistant." Already, some buying agents and comparison-shopper agents are accessible to many people (e.g., eBay) over the Web.

It is fairly easy to imagine some kind of agent-based calendar assistant or e-mail assistant very soon for those who use PCs, Jornadas or Palms. It is also easy to see simple agents appearing in cable set-top boxes, just as the VCR is being displaced by "intelligent" video recorders with learned preferences.

It's a bit further off if we want agents to support buying from phones, or more intelligent interaction through speech, or more collaboration between the personal assistants of multiple people ... then each person in the interacting community needs to have one, they always need to be running, they need to be secure, etc.

How does your work with software reuse tie into your work with agents?
Component reuse is enhanced when components are loosely coupled, flexible and adaptable because components can more easily be combined into novel, robust combinations. So, agent technology is a way of making very flexible reusable components. To some degree, an agent component adjusts itself to its environment, making it more reusable.

www.cooltown.com/impulse

"Recently FIPA has established a set of standards, which many of the newer agent systems try to follow. In particular, there is a standard for an agent communication language (ACL) and the model of a standard agent platform and agent services, which is intended to make it easier for separate agent systems to communicate."