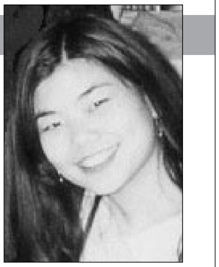


By Crystal Chweh
Editorial Assistant
cchweh@computer.org

Generations ahead: Michael Huhns on cooperative information systems

"Cooperative information systems are the future of how information systems will be constructed and how they'll operate," says Michael Huhns, director of the Center for Information Technology at the University of South Carolina. They are, he claims, "two generations ahead of the current state of the art."

Cooperative information systems can be defined technically as "a combination of databases and agents." (See Figure 1.) Ranging from databases for chemists to healthcare information systems to environmental databases, CIS is "an appropriate architecture for all information systems of a reasonably large size. The computer that runs your household doesn't need this technology, but if you have a business with several computers or larger than that, this is an appropriate technology."

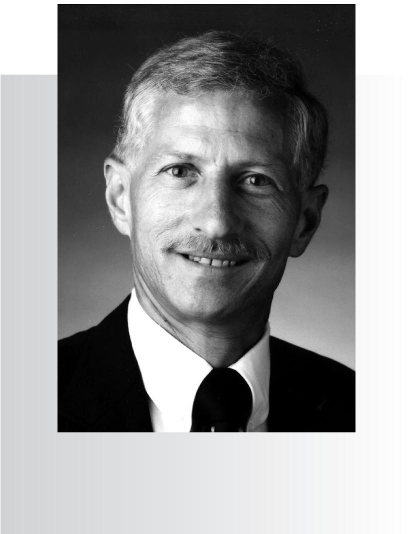
Breaking it down

Huhns categorizes the development of cooperative information systems into four generations. First are centralized information systems. "Originally, everything was centralized. Most organizations had one computer, with the databases all in one place. Now information is distributed, but not necessarily very coherent or very organized."

Client-server architectures are the second and current generation. "The current paradigm for information systems is based on a client-server architecture," says Huhns. One of the drawbacks of client-server architectures stems from the passive nature of the servers (primarily thought of as database components). "Servers are idle until a client asks for something. If a request is made, but the server receives new, relevant information a second too late, then it doesn't do anything; it has no obligation to inform the client."

Third are peer-to-peer architectures, in which clients communicate directly with other clients, and servers with other servers. "They're distributed systems, and more active, but not necessarily cooperative. The key lies in adding cooperation to distributed systems; this "achieves the next level of interoperation."

Cooperative systems are fourth. "Cooperative information systems treat each component as being active and cooperative. So if a database learns something new, it can update and inform the user, even long after the initial query. The server becomes a more active participant and can work with other servers (or their equivalent) to gather information for the user."



Pulling it all together

Huhns, formerly a senior member of the research division at the Microelectronics and Computer Technology Corporation, has been working in cooperative information systems in the foundation areas of agent technology, from the early eighties. "The development of cooperative information systems is occurring on four levels. The guest editors did a good job of describing and characterizing those four levels: interoperation, coordination, information services, and change management" (see "Cooperative Information Systems," pp. 32–35). Research is going on at all of these levels, and some of the major problems they encounter "involve making sure that the systems can scale up," says Huhns. "As you hook many information-system components together to cooperate, there will be a lot of communication, or network traffic. You don't want them to spend all their time chatting and getting nothing done. A lot of effort is involved in creating efficient algorithms by which groups of components can cooperate and work independently, but coherently, toward a common goal, and resolve conflicts and inconsistencies."

Years of research and work in CIS have given Huhns a unique perspective. "I've had research projects at most of these levels. I've developed agent-communication languages at the interoperation level, as well as semantic repositories or ontologies. In the next level up, coordination, I've developed some multiagent protocols for cooperation, and techniques for enabling several agents to work together cooperatively. At the information-services level, I've produced applications that achieve information retrieval across a variety of resources, or that manage workflow in a heterogeneous database environment."

"The hardest problems are at the higher level of this layout, change management," says Huhns. He notes, though, that the problems aren't always within the technology. "This level requires more understanding of the semantics of information systems. Currently, people provide this understanding. I would like to automate many of these things where possible, enabling larger, more robust systems to be constructed. But it's a hard problem—it requires agents that are much more intelligent than the ones we can construct now. So, a motivation for the field involves trying to develop large systems, but have each piece be relatively small and simple, and in many cases, the same piece that it's always been. We're trying to build large systems out of the smaller, more manageable systems and more easily understood components."

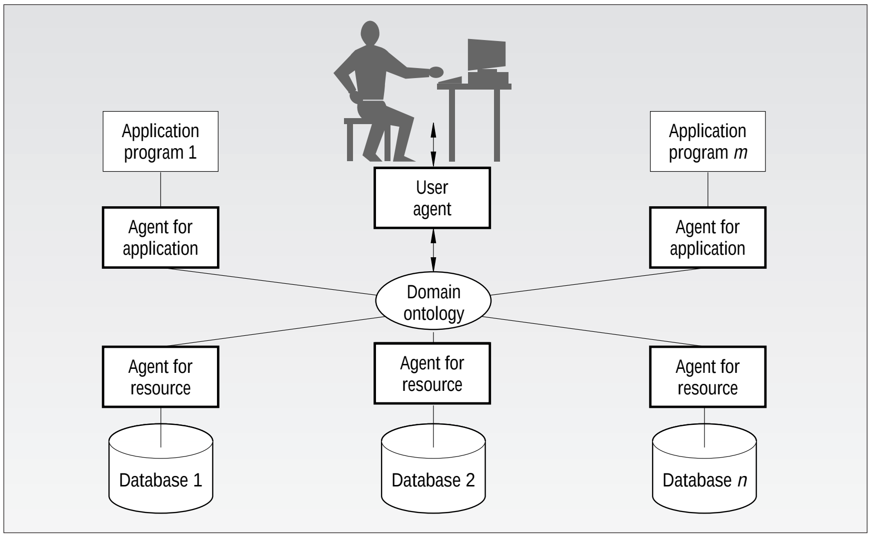


Figure 1. In a cooperative information system, any database system, application program, user interface, or agent can communicate with any other.

Communication is the key

"From a more traditional database standpoint, the current action or the hottest topics are in managing transactions (updates, or making changes) across heterogeneous components," asserts Huhns. "In particular, research in this area is to develop relaxed-transaction processing. If you have a database, one half of the problem is getting information out and the other half is putting new information in. If you have several databases, you might want to update all of them, but you want to make sure the updates are all consistent. Problems arise if there are a lot of databases and they're distributed and not all under your direct control—one might be busy doing something else. You can update one and if the other doesn't get updated, they're inconsistent. One system might be down for a while or a transaction might fail, so one half of the update finished, and the other half didn't. Now what do you? Possibilities are to undo the one that succeeded, try the failed one again, or try something else."

"There are procedures in the database community for doing this when you have a small, fixed set of these servers under your control. One of these procedures is called a two-phase commit protocol. This protocol gathers the full attention of the databases and forces them to proceed in lock step until the transaction either succeeds completely or fails completely. This doesn't work if the number of servers is large, they're very distributed, or they're not under your control. So, new techniques are needed. These are typically called relaxed-transactions techniques, because some of the consistency requirements are relaxed. There are a number of proposed solutions being evaluated now."

"One other area where there's a lot of activity involves the development of ontologies, or conceptual models of some domain. If there are a number of different services or sources or applications, and they're built independently, then they'll have different semantics, terminology, languages, and representations, but they all need to work together. It would be nice if from one place, you could understand everything that's in all these heterogeneous systems. An approach is to develop a single common model that describes all of them—that model is typically called an ontology. If you can construct one of those, then you can use it as a basis for querying or updating the various systems; understanding, managing, and mediating between these systems; and enabling them to interoperate. The ontology becomes an interlingua, an Esperanto."

Bottom line: it works

For the most part, CIS is a research area with prototype systems in a few environments and a few applications, but it's got a bright future because of agents. "The world has a trillion or more invested in information systems," says Huhns. "No one's going to throw away their old information systems and build a cooperative one. Well, new systems can be built using CIS technology, but they'll be glued into the existing legacy sytems using the same CIS technology. There will be agents that represent both the old and new components. For example, old databases might be the network type, with a procedural query language, but there'll be an agent to translate a relational query language into that procedural one. The procedural language might be very arcane, but the agent will take care of that. If you were to build the system all over again, you'd use a modern relational database that directly understands a relational query language. Its agent wouldn't have to do any translation. From the outside, both of the systems would appear to be the same: they could both process relational queries. The new one would be easier to manage, have better tools, and might be more efficient. The old one would have the advantage of an old proven system that works—so let's leave it alone. It's a way of evolving into newer systems."

Cooperative information systems can save businesses currently running legacy databases the cost of entirely replacing them with new systems. "To do that, take old components and an active piece of software—an agent—that will help them be cooperative. Agents will represent each of these components and, even if they're passive, the agent will make them appear active. Even though they're very different, the agents will make them appear homogeneous. This technique will allow old existing systems to evolve into cooperative ones, without having to replace them," asserts Huhns. "The essence of this architecture is having all the different components being able to work together. Agents can make this happen; these old components will fit in naturally until they can be replaced, so there's a nice migration path."

Cutting through the red tape

As a founder and board member of the International Foundation for Cooperative

Information Systems, as well as being on the editorial boards of several respected journals and magazines (including *IEEE Expert*), Huhns is known worldwide as an authority in the field. One project that helped bring him such acclaim was Carnot, "a project in providing interoperation among heterogeneous distributed databases." On this MCC project, Huhns worked on a team to produce "ontologies that described the domains under consideration. The databases were heterogeneous; some were object-oriented, hierarchical, or relational; they had different schemas and different query languages. We made them all work together. There were small agents that formed the glue or the middleware that made these systems interoperate." The success of the project led to its application at Eastman Chemical. "They had a number of databases that chemists needed to access. Chemists, in the past, would talk to a database analyst who would write queries for them. Getting information took from a week to a month; our system enabled the chemists to get the information themselves, in just a few minutes."

Another Carnot-based application was implemented at Ameritech to help manage a workflow that they had in an area called service provisioning. "Prior to our automating this," says Huhns, "the workflow took approximately two weeks to complete; afterwards, it took approximately two hours to complete." The major benefits realized by these applications led to a follow-on project called InfoSleuth, also done in conjunction with MCC. "This is a set of heterogeneous agents used for information retrieval." The project's goal is to expand application beyond distributed systems into a more dynamic environment, such as the Internet.

What's the future of CIS?

Huhns got involved in cooperative information systems because it "seemed a fertile area for research." As a professor of electrical and computer engineering at the University of South Carolina, he's in a good position to influence others to take an interest. "This current semester, I'm teaching a graduate course in cooperative information systems," says Huhns. "Students seem to appreciate it. They feel there're learning something that will stand them in good stead for several years." He's managed to convince them that the "problems here are very important, because information systems run the world's businesses." Along with the practical applications, some of the basic problems in CIS address theoretical issues such as "the theories of database transactions and cognitive theories of how agents should reason and how they can reason rationally." Huhns has done work in truth-maintenance systems that demonstrates the "theoretical underpinnings for how agents can maintain logical consistency." "There are many great theoretical problems involved here," argues Huhns, "but there are nice, practical reasons for working on those problems."

In cases where his students need further motivation, Huhns describes his current projects: "I have current research projects in helping to unify information systems across different government agencies. One program involves environmental databases that are maintained by the US Environmental Protection Agency, the Department of Energy, the Department of Defense, and the European Environmental Agency. They all have environmental information, and they want it to be accessible to their employees who need it, as well as to the general public." Right now, accessing this information would require extensive knowledge about each individual agency, as well as the means to access the different systems. "We're providing what appears to be one unified system, even though the different systems will still be distributed and autonomous. Users can get to all the information from a single place, with a single request, and it'll all be translated into their own terminology, independent of the terminology of the individual systems."

Huhns is also working on similar projects ranging from logistics information maintained by the Department of Defense to health-care information systems that provide interconnections between systems in hospitals, practitioner's offices, health-maintenance organizations, insurance companies, and government agencies. "These current projects have been underway for several years; in each case, they'll first be a demonstration or prototype system. We hope from that, they'll be picked up, spread, and commercialized, and the technology will be more widely used." Cooperative information systems has an ambitious and wide-ranging field of applications. With characteristic enthusiasm for his work, Huhns asserts, "I think that this is an extremely promising area—the predominant paradigm in the future."

How to Reach Us

Writers

For detailed information on submitting articles, write for our Editorial Guidelines (s.bergen@computer.org), or access http://computer.org/expert/edguide.htm.

Letters to the Editor

Send letters to

Managing Editor
IEEE Expert
10662 Los Vaqueros Circle
Los Alamitos, CA 90720

Please provide an e-mail address or daytime phone number with your letter.

On the Web

Access http://computer.org/expert/ for information about *IEEE Expert*.

Subscription Change of Address

Send change-of-address requests for magazine subscriptions to address.change@ieee.org. Be sure to specify *IEEE Expert*.

Membership Change of Address

Send change-of-address requests for the membership directory to directory.updates@computer.org.

Missing or Damaged Copies

If you are missing an issue or you received a damaged copy, contact membership@computer.org.

Reprints of Articles

For price information or to order reprints, send e-mail to s.bergen@computer.org or fax (714) 821-4010.

Reprint Permission

To obtain permission to reprint an article, contact William Hagen, IEEE Copyrights and Trademarks Manager, at whagen@ieee.org.

IEEE EXPERT
INTELLIGENT SYSTEMS
& THEIR APPLICATIONS