

A different kind of Internet

It's been just more than 40 years since Robert Kahn took a leave of absence from the Massachusetts Institute of Technology faculty and helped create a computer networking project whose effects are still being felt today. His design ideas enabled the creation of ARPAnet, the world's first packet-switched network, and ultimately led to the Internet. His subsequent work at the Defense Advanced Research Projects Agency, where he led the Information Processing Techniques Office, would lead to the largest computer research and development program ever undertaken by the federal government.

As chairman, president and chief executive officer of the nonprofit Corporation for National Research Initiatives, Kahn continues to be a global advocate for long-range infrastructure research and open-architecture networking. He still harbors a vision for how the Internet could be used to manage information, not just move packets of information. He shared that vision in a recent interview with GCN Editor-in-Chief Wyatt Kash and explains how it could help make electronic health records and other data more secure and permanently accessible.

GCN: You started CNRI to foster experimental research projects that are national and infrastructural in nature.

Where are you focusing your efforts these days?

ROBERT KAHN:

Interestingly, a lot of the stimulus focus is on infrastructure creation and improvement. Some are of the traditional kind, where you need jackhammers and concrete. But some of it is informational — in areas like getting medical records online. We have witnessed explosive growth over the last two decades in a variety of computation capabilities and the Internet in particular.

We try and come up with good system and architectural ideas such that those kind of projects can be carried out successfully on a national scale.

GCN: What has evolved from your work around Digital Object Architecture and the Handle System?

Well, the Handle System is one component of the more comprehensive architecture that I call the Digital Object Architecture that was intended to reflect what the Internet might have looked like if its main goal was to manage information, as opposed to just moving bits or packets

from place to place.

The key element of the architecture is the “digital object,” or structured information that incorporates a unique identifier and which can be parsed by any machine that knows how digital objects are structured. So I can take a digital object and store it on this machine, move it somewhere else, or preserve it for a long time by porting it from place to place.

A digital object doesn't become a digital object any more than a file becomes a file if it doesn't have the equivalent of a name and an ability to access it. The fact that you type a few keys on your keyboard

doesn't make the typed characters a file. You have to give it a file name and put the characters into the file system. So these digital objects have identifiers called digital object identifiers — or “handles.”

Say the identifier for this object is 1234/XJ493267. There needs to be a resolution system that you can ask about the digital object that has this identifier. For example, what are the IP addresses where the digital object can be accessed? So the Handle System resolves these handles, once submitted over the 'Net, into handle records — and it gives your computer the handle record for that identifier almost instantly.

Now, what this rapid resolution capability can do for managing information depends on what the user puts into the handle record. For example, he could put in the many places on the Internet that this object is stored; it doesn't have to be just one. He could put in authentication information for that object, so that you could verify that the object is what it purports to be and that it wasn't corrupted along the way. It could be public keys for access. It could provide various terms and conditions for use, or give you some sense of what you might do with this object.

Another part of the architecture is the notion of a repository, where digital objects may

be deposited and from which they may be accessed later on. Repositories make use of existing storage systems.

The Handle System might provide the IP address of the repository. You then go to that repository and say, “I'd like to access the object whose identifier is X.” With many existing database systems, you've got to know a lot of the details to get in. But the repository software can totally insulate you from all of these details. It makes it easy to store a digital object, easy to access a digital object (or parts thereof), and you can use the technology to preserve the object for a long period of time. You never have to worry about redoing the digital object when you change from one system to another, whether it's the same manufacturer or a different one. You just port the digital object from place to place.

ZAHID HAMID

Now, if you've encrypted that digital object, you may not be able to find it if you didn't keep very good records of its identifier. So, enter the notion of the metadata registry, which is a system that you can interact with that will help you locate the digital objects when you don't know the identifiers to start with. It will also let you build collections and identify collections of digital objects that might span across many different repositories. So it's a really powerful kind of capability. Parts of the Digital Object Architecture are widely used by the publishing industry.

How does the architecture allow the registry to stay current?

Remember, the Handle System is a big, distributed system. It's not a single server in a single location. It consists of many servers (or services, actually) running in lots of different places running local handle services, each of which is itself potentially distributed.

Suppose there are lots of different users that have copies of electronic journals. Let's say you don't even know who they are. When one of these users tries to access a reference cited in some other electronic journal and clicks on it, his system first goes to the Handle System to pull back the handle record. If the Handle System is updated when a change occurs, every one of the users everywhere suddenly will have access to the current information.

The system uses a set of procedures that define handles as having a prefix, a slash and a suffix. The suffix can be anything you like: an existing name, a numerical sequence, a Social Security number – it could even be a cryptographic

sequence. The prefix is what distinguishes it. The prefix is given to an organization or individual so that when they create the suffixes, the handle is guaranteed to be unique. You could operate a local handle service yourself or use a service provided by someone else.

When you install a local handle service, it first creates (locally) something called a site bundle, which contains information like the IP address where it's located and a public/private key pair. You keep the private key, and send the



Digital Object Architecture is almost ideal for what government and the health sector need for handling medical records online.

public key to the administrator of the Global Handle Registry. That administrator will then allot a prefix to the local handle service, enter it into the global registry on your behalf and enable you to then change the handle record entry in the Global Handle Registry using your public/private key pair.

For example, a university might have an allotted prefix, say 1500. They could allot 1500.1 to some entity within the university, which in turn, can create 1500.1.A, or 1500.1.B. The university can also create a prefix with semantics, say, 1500.headquarters. And they can create these derived prefixes all on their own; they don't have

to work through anybody else. Every one of these new prefixes can be separately administered under its own public/private key pair. It's really pretty powerful.

Why hasn't the architecture gained greater traction?

The original idea for the digital object was mine; it grew out of some work that I had done with Vint Cerf back in the mid-1980s on what we called knowledge robots, or Knowbots programs, which are mobile programs in the Internet. The details of the

Handle System were worked out by one of our best programmers at the time, David Ely. And over the years, many other people have played a role in helping to create parts of the Digital Object Architecture. We continue to work on evolving the components of the technology and applying it in different contexts.

But you need to work with people to help them apply it. We're working on the applications of this technology, and injecting in it to different applications. We also haven't really done much marketing of the technology, but this could change.

Where might these applications evolve — especially for use in government?

One area that this is clearly an excellent application is archiving. Just about every organization in government has to retain information. The problem is that most parts of the government deal with archiving in a limited context using readily available commercial technology or services. So if you retain information on a big external disk and need something, you retrieve the disk and plug it in. Sooner or later, you're probably going to have lots of these disks, and need to access a piece of information when it's not online, and you can't really get your hands on it.

We have experimented with some archiving capabilities on the 'Net, some developed by CNRI and some by others, that are intended to serve long-term archival storage needs.

My hope is that we can make the digital object technology, which operated in the Internet environment, available as we did with the original Internet technology and get a lot of people in the public and private sectors to understand its power and capability. Because it's an open architecture, it has the potential to grow organically as did the nascent Internet.

Another application area that I'm very optimistic about is medical informatics. Digital Object Architecture is almost ideal for what government and the health sector need for handling medical records online. But the bigger issue here, as it has been for years, is privacy and how you deal with that requirement across different organizational boundaries.

ZAIQ HAMMID

The Web as originally deployed assumed that everything would be publicly available. But over time, Web sites have cordoned off parts of that public place. For example, you may need passwords to get into this Web page and access that information. And you have to know exactly where to look for certain information. That's where search engines come into play, assuming they can index the information you want from public spaces or by private arrangements.

The Digital Object Architecture was designed, at least at the level of repositories, exactly from the opposite point of view. Namely, it assumed that what you're storing is your own information, and it's not available to anybody else.

So the protection occurs at the object level rather than with protecting the identifier or by providing only a password at the boundary.

How do you deal with the need to authenticate that someone is who he says he is?

By putting this information in a public-key infrastructure, such as the Handle System enables, the technical means is available to verify that the user is the person who holds the private key corresponding to the public key held in the system. In the repository world, every individual would have a unique persistent identifier which they can use to verify that they are who they purport to be. During the initial transaction with a repository, the repository would ask the user to encrypt a random string with his private key, send it back so the repository can verify the user. This same technique can be used (in

reverse) by the user to verify that the repository is who it purports to be. Of course, if a private key is lost, that information needs to be made known to the system as soon as possible so that public/private key pair may be revoked.

This is only a start; it's the infrastructure part. Ultimately most of the effort goes into deciding what information you want to keep and not keep. If you keep it, how do you want to characterize it? When you identify things, how do you want to do the identification?

The Web assumed that everything would be publicly available. The Digital Object Architecture was designed...from the opposite point of view. So the protection occurs at the object level, rather than with protecting the identifier or by providing only a password at the boundary.

And who do you want to have on what lists, and who is going to manage the movement of the information when you eventually decide to move it from one place to another? And when you create new types of information, can the metadata for that information be extracted automatically, or will individuals continue to play a role in generating and maintaining it?

What has become of your work with Knowbots?

We use them in a number of different contexts and in many applications. I think the reason that it didn't get as much adoption as I thought it could was because there was a con-

cern on the part of many organizations and their IT staff that these mobile programs should be treated like viruses. I thought this was unfortunate because it should have been possible for them to validate the kind of operations that a Knowbot Service Station, which runs the Knowbot programs, could take and delineate a set of actions that could be invoked in a Knowbot operating environment.

It would be the equivalent of the old automats in New York

along to another location and eventually return to us a single consolidated view of all the collected information.

The infrastructural capabilities that we've created are available to the research community. They enable somebody to program them to do a specific task, but they don't come fully programmed for any specific task

What other projects might be of interest to the government IT sector?

We're involved in a number of other efforts. Clearly, we're interested in the whole field of networking. I've been involved in a lot of the international deliberations concerning the future of the Internet. We are planning to spend more time on issues of medical informatics, because we have technology and system concepts to contribute there. We also plan to continue working in the archiving area. We're very interested in how people manage collections of information, so we've been doing some work with a Pentagon project called Advanced Distributed Learning. Finally, we're also involved in fostering research and providing prototyping services for the fields of micro-electro-mechanical systems and nanotechnology, using an approach similar in some ways, although different in many others, to one that we developed with the research community when I was working at DARPA.

And we constantly are looking for ways in which we can help with new national and even global infrastructure initiatives.

(See more of Kahn's comments at GCN.com/1333).