Lt Col Doug Dyer
Information Awareness Office (IAO)
Genisys

Good morning. I'd like to tell you about a significant information technology problem that I think you'll all understand, about its root cause, its impact on our ability to combat terrorism, and what DARPA is doing about it. It turns out we're not getting anywhere near the full potential from our computers. Today, we use computers in the wrong way, namely to format, store, and send information . . . but the work is all done by people, and products are consumed by people. In fact, most digital information consists of natural language, sounds, graphics, images, and other formats that can only be interpreted by humans. Is it any wonder we can't get our computers to provide more automation, be smarter in dialog, or offer any kind of intelligent assistance?

To work, algorithms need structured data; that is, variables with values. Yet, if you analyze your own computer today, you'll probably find that of all the valuable data on your hard drive, less than 1 percent exists in any structured form.

Lack of automation and the structured data required for automation is absolutely not a new problem. In fact, DARPA and other organizations have been researching ways to make algorithms understand natural language and to add semantic markup to web pages so they may be interpreted by software agents, for example. However, the quintessential source of structured information has been and is the database. This morning, I'd like to explain why current databases are inadequate, describe why databases are essential to combating terrorism, and tell you about a new DARPA program called Genisys aimed at totally re-inventing database technology.

Relational database technology in use today was created in the mid-1970s when processors were slow, networks were created with modems, and disks were tiny in comparison to those we have now. As a result, database implementations stressed time and space efficiencies at the expense of flexibility and ease of use. Although current databases are relatively straightforward, they require careful design of data structures before they are populated, and both users and application programs have to know a great deal about the design in order to access information.

In addition, once the design of data structures is complete, it's difficult to make changes because interfaces used by applications must be rewritten for the new design. Furthermore, no conventions are used when creating terms to name real-world entities and their attributes. Terms are invented ad hoc by database designers and, as a result, people who are new to the database have some difficulty interpreting the information it contains. For these reasons, it's more difficult than it needs to be to store, share, and use information in a database, and this has stymied innovation, in genera,l and the creation of large-scale automation and intelligent processing algorithms, in particular. You may have personal experience creating a database—if you have Microsoft Office installed you already have a relational database called Microsoft Access—even for simple projects, most people find relational databases difficult to use.

Lack of intelligent automation and databases that enable automation hurts every one of us who wants greater efficiency in our day-to-day business, but it's a particularly acute problem for combating terrorism. There are many terrorist organizations, and they each have many targets from which to choose. Combating terrorism requires us to sift through gigabytes of information, recognize patterns, and share information among organizations. Doing this quickly enough to matter requires automation and the databases that support it.

By analyzing a number of terrorist attacks, it's clear that having timely information is key. The sooner you know about an attack, the higher the probability that you will be able to preempt or mitigate it—if necessary, to respond and recover and, if possible, prosecute or take appropriate military action against the terrorists. Fast, certain response is one of the best methods of deterring future attacks by other terrorists because it raises the possibility that their actions will result only in failure, humiliation, and punishment.

So, how do we get this information? It turns out that before every attack, critical events occur. Supplies must be purchased, weapons may be developed, rehearsals are conducted, and elements must be positioned. In almost every case, separate "systems," if we use that term loosely, capture partial evidence about these events.

Examples include records of purchases and other transactions, messages and communiques, facilities and ownership, travel itineraries, and relationships that can be inferred. These events often are reconstructed after the fact to support criminal prosecution, but we'd really like to be able to compile the information before an attack, rather than afterward. Unfortunately, we aren't tapped into these systems. Our intelligence organizations are not optimized for terrorist threats, so we'll need to build new systems and populate new databases to get the coverage we now need. An example helps to illustrate:

In 1995, Aum Shinri Kyo attacked the Tokyo subway system using sarin gas, killing 11 and sending hundreds to the hospital. This attack is a prototypical, perhaps extreme example of predictive precursor events. Prior to the 1995 attack, Aum Shinri Kyo cultists, led by Shoko Asahara, had tried to buy U.S. and Russian chemical munitions. When that failed, they engaged in a substantial weapons development program aimed at mass effects. They created elaborate development facilities for producing sarin, purchased expensive equipment, and accumulated huge chemical stockpiles. Following several malicious attempts with ineffective agents, they created a test facility in Australia and tested sarin by killing sheep, whose remains were later discovered. Noxious chemical leaks were reported by neighbors near their development facility in Japan, but police still made no arrests. Asahara broadcast his intentions clearly by radio. And months before the subway attack, cultists used sarin in an overt attempt to kill three judges in the city of Matsumoto. In this example, just as in the case of 9/11, fragments of information, known by different parties in different organizations, contained a predictive pattern that could have been identified had the information been shared and properly analyzed.

To address these issues, we've created the Genisys Program. Genisys has three goals: First, we'd like to be able to integrate and, if desirable, restructure legacy databases. Second, we want to dramatically increase the coverage of vital information by making it easy to create new databases and attach new information feeds automatically. This is new, multimedia, broad-spectrum information that doesn't exist in any structured database. Third, we want to create brand new database technology based on simple, scalable, distributed information stores we call repositories. In contrast to today's databases, repositories will be able to represent a broad array of information that varies in terms of structure, certainty, and format, and accessing information will be easier. Operationally, we will focus Genisys on the problem of combating terrorism. In the context of a larger counterterrorism information system, Genisys repositories will both supply and receive information. Initially, repositories will be populated with synthetic data to support experimentation and rapid prototyping, but our intention is to iteratively develop and transition the technology, using operational feedback to focus future research. This is a model we call "assured transition." Now I'd like to describe in technical detail two of these program goals: database integration and the core repository technology.

There are three well-known methods of integrating databases today. As applications are built, new interfaces can be created as needed. As the number of applications and databases increase, this method requires a quadratically increasing number of interfaces, limiting scalablity. Alternatively, a software agent known as a query mediator may be used to translate application queries into queries that are understood by different databases. This approach improves scalability somewhat, but often the complexity is simply moved into the agent. A third approach involves manual reengineering of the relevant databases and subsequent manual re-implementation of the interfaces. For mature systems, this is sometimes worthwhile, but it involves the greatest amount of reengineering and, hence, the greatest cost. A primary technical limitation shared by all these approaches is the lack of design conventions for creating terms that name real-world entities and their attributes.

A new technical approach identified early in Genisys is to use domain ontologies, which are the terms and relationships that are associated with the concepts in a particular model of the world. Once there is an ontology, it's possible to define a naming convention and reduce the cost of designing the database. It's also possible to create a database crawler, similar to a web crawler, that discovers the structural design, or

schema, of a legacy database. Using the existing schema and ontologies, it is possible to create tools that help people map the old schema into the new one, reducing the cost of reengineering and integrating databases dramatically. It's also possible to create a simpler query language than today's SQL and, perhaps, automated tools for translating interfaces. Using this approach, we believe legacy databases can be transformed and integrated cheaply. Now, let's look at the design criteria and goals for new repository technology we are defining.

Today, processors are a commodity, high-speed networks are ubiquitous, and we're beginning to think of disk space as infinite. Today's infrastructure suggests new design criteria that will result in repositories that are friendlier and more capable than current databases. For example, current databases support analytical queries very well, but they have several drawbacks that make them inflexible and hard to use.

First, they require a priori data modeling. That is, the database must be designed before it can be populated. Moreover, once designed and applications begin to use it, changes to the design are difficult because they require modifications to the interfaces for applications. Repository technology will change this by making applications independent of the physical structure of the database. For example, queries will no longer refer to table names in a relational implementation.

To provide even more flexibility, we want to exploit search technology so applications do not even need to know the physical location of data. These two improvements will make creating and using structured data easier. To enhance sharing, we're interested in intelligent processing to help enforce policy while automatically granting access to people who should have it. We also want machine assistance for declassifying data to release the sharable content while protecting the sensitive parts. Inside the repository, we want to be able to represent uncertainty in some natural way. Finally, we want automation for restructuring the content to increase performance, add new information types, or solve new problems. Clearly, this is a long list of goals, and we may need to trade off some things to achieve others. Of paramount importance are scalability and ease f use because, ultimately, we want to a lot of structured content relevant to combating terrorism.

Specifically to address the privacy concerns of those not connected with terrorism, repositories need technologies for protecting personal privacy. We have three methods relevant to protecting privacy. First, we can exploit partitioning to separate identity information from transactions that people conduct, only reforming this association when we have evidence and legal authority to do so. We can also use partitioning to project specific information authorized for the particular role of the requestor. In this way, the Center for Disease Control, for example, could access recent statistical medical information while others could not.

Second, we will develop and employ information filters to keep information that is not relevant out of the repository. Filters could be used to implement laws and policy that regulate the kinds of information recorded and who it pertains to. Finally, we will use software agents to mine the information in the repository, form associations from content, and expunge information found to be unrelated to combating terrorism. Filters are not perfect, and this last method will help ensure privacy when filters fail or when the combination of bits of information make it clear that the information has no utility. These three method are initial steps toward achieving personal privacy and security. They are probably inadequate, and we are soliciting additional new ideas in this area.

We are just getting started in Genisys and don't yet have tested designs that could implement these characteristics, but we believe our goals are attainable. With Genisys, we'll be able to reduce the cost of integrating databases so different DoD and other Government agencies will be inherently better at coordinating.

In addition, increasing coverage from many different sources by autopopulating large information repositories means that we're much more likely to have the information we need to prevent or respond to an attack. Other programs at DARPA will use content in the repository to match patterns and support analysis and, thus, we'll have a higher probability of recognizing attacks. The goal for Genisys technology is to make it cheap and simple to build these autopopulation and reasoning systems, while still accommodating scale, representing uncertainty, projecting data for sharing, and helping to manage the information automatically.

In March, we published a Broad Area Announcement, DARPA BAA 02-08, and have made an initial round of proposal selections. We're in the process of getting selected offerors under contract. We welcome your good ideas and support. At this point in the program, there are still opportunities to join the team for anyone with the potential for outstanding contributions.

Thank you.