

## RESOURCE DESCRIPTION FRAMEWORK DATABASES

### INTERNET ENABLED DATABASES FOR TEST DATA MANAGEMENT

#### Comparing Data Models

During every waking hour, people sort, file, and compare the objects we see around us. We draw boundaries around collections of things, creating higher-level data categories. Consider, for example, a cloud, an area where a gaseous solution is in a certain state. A cloud is not really a single object, yet we have the ability to treat it that way. This ability—a hallmark of the human brain—adds an immense richness to the ideas we are capable of expressing.

When dealing with highly abstract information, the relational data model has stood the test of time. The relational model is the basis for relational databases. As the name suggests, it describes the relationships between information. The allowable relationships must be defined in the database's schema before any information is entered. Adding new fields to an existing database requires all the records to be updated. Querying the database also requires existing knowledge of the schema, and new relationships cannot be discovered during a query.

For example, let's look at a table of test records. Each has an associated ID, part number, procedure document, serial number, results document, test outcome, and date/time. Because these relationships are known and unchanging, we can easily describe and query them in a relational database. A simple query might be to find all test records that failed between two dates. Relational databases excel at this type of challenge. They even allow tables to hold references to entries in other tables. When the database is queried, the tabular information is joined to express a result.

However, performing a large number of joins slows the response to a query. All relational database implementations share this "join problem". In addition, if we want to add more columns for other tests and their outcomes, we must update every test record in the table.

In contrast to the relational model, the directed-graph model assumes very little about the data it holds. Each piece of information is connected to others via an arbitrary relationship. The relationship can apply from one piece of data specifically to another, or it can operate in both directions.

ID	Part_#	Procedure_Doc	Serial_#	Results_Doc	Outcome	Date_Time
0	41224	41224_Test1	1001	051503_120117_Test1	Fail	5/15/2003 12:01
1	41224	41224_Test1	1001	051503_120231_Test1	Pass	5/15/2003 12:02
2	41224	41224_Test1	1002	051503_120321_Test1	Pass	5/15/2003 12:03
3	50917	50917_Test1a	253	051503_113211_Test1	Pass	5/15/2003 23:32
4	50917	50917_Test1a	254	051503_113211_Test1	Incomplete	5/15/2003 12:01

The following examples show how a directed-graph database would store the data from the first test record in our sample relational table (Figure 1).

To add more data for other tests and their outcomes, we would simply create two new objects that are linked to the original test record object (Figure 2). Other test records do not need to include similar information, so no additional database maintenance is required.

There is no limit to the number of relationships that two pieces of data may share, nor how much other data may be connected. The data and relationships form an interconnected web, eliminating the need for joining complex tables.

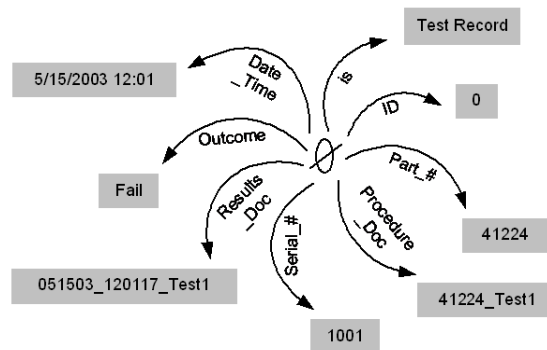


Figure 1

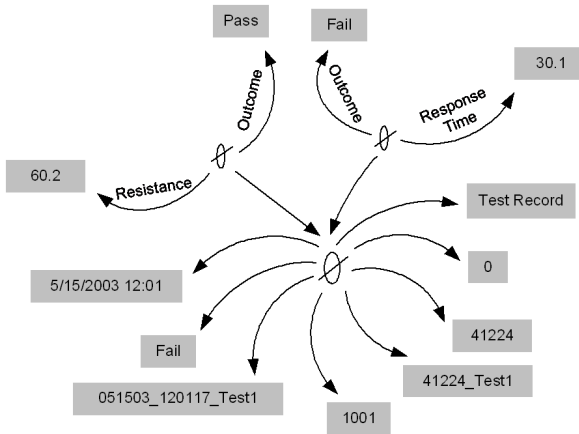


Figure 2

A directed-graph database allows the freedom to build user interfaces that have no prior knowledge of the data's schema—something impossible in the relational world. The system can grow with new sources of data and new or changing test requirements, without changing the underlying structure of the database. Reducing the need for database maintenance dramatically reduces the long-term cost.

These concepts can be extended to handle sophisticated information relationships. Another use for this technology is the creation of a Device History Record database for cradle-to-grave record retention. If we add a serial number object and attach significant document links to it, we have effectively created a management system for all documents pertaining to a serial number. The storage mechanism for these documents does not have to be part of the database—only the links (Figure 3).

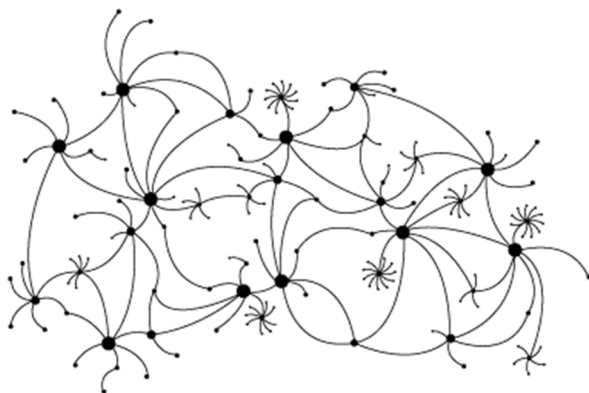


Figure 3

### The Tucana Knowledge Server

The Tucana Knowledge Server (TKS), from Tucana Technologies, is a type of database that natively implements a directed-graph model. It uses item-property-value statements, with each statement containing two data elements (the item and the value) and a relationship between them (the property). All information enters the database in this standard form, the Resource Description Framework developed by the World Wide Web Consortium.

Sorting all statements by item, again by property, and then again by value effectively flattens the data structure. Any query constraint can be satisfied by a simple range in one of the indices. Though somewhat less efficient in the use of storage, this approach delivers dramatic improvements in performance because it is optimized for joins. Query times in relational databases can grow exponentially with the inter-relatedness of the data. Query times in the TKS grow roughly linearly in proportion to the amount of data.

**RDF databases can reduce query times.**

Directed-graph databases are not a universal replacement for relational databases, which remain the best tool for environments where complex queries must be run against structured and relatively simple data. Many real-world problems are best solved by using more than one approach. Directed-graph databases, however, clearly excel at handling large volumes of distributed metadata.

### Storing and Retrieving Metadata

The TKS is optimized for the storage and retrieval of metadata, or information about data. The process of metadata management consists of complex queries performed against highly interrelated information (metadata statements). This makes metadata management a good application area for a directed-graph database.

**The graph may be easily "walked" to discover new relationships among the data.**

TKS works by representing large amounts of information with a (typically) smaller amount of metadata. In the case of test records, metadata might include the operator, the fixture, the last calibration date, repair or rework information, notes, etc.

Each test record might have a different set of metadata, depending on the model tested and the procedures in place at the time the test record was produced. This ability to associate different metadata to each item provides the flexibility to withstand changing requirements. The more traditional approaches have proven to be difficult and expensive to maintain as requirements and procedures have changed over time.

Metadata is stored in the Resource Description Framework (RDF), which enables sharing across applications and geographic boundaries. RDF is part of the Semantic Web project, an important next-generation project to facilitate the understanding of information by the machines that store it.

**The Resource Description Framework (RDF) is an international standard.**

Original data is stored separately from the TKS and is referenced by a uniform resource locator (URL). This data may be stored on a simple file system since it is accessed by machine. This avoids storing large documents in relational databases or object stores, and it drastically speeds access to the underlying information.

The Semantic Web project has seen the creation of other RDF databases that have been useful only to academics pursuing RDF research. Of the commercial, scalable products available, only the TKS was created from the ground up as a native RDF database and is not based on existing relational database technology. This is important to the scalability of the TKS, the first commercially available building block of the Semantic Web that can be used in large-scale real-world applications.

**Stored separately, the original data is protected.**

## Conclusions

A directed-graph data model should be used where there are complex and rapidly changing relationships among the data objects. The Tucana Knowledge Server implements a distributed, secure, directed-graph model of metadata. As the urgency of gaining control and exploiting the strategic value of unstructured information continues to grow, enterprises will find more and more business-critical uses for RDF databases.

In summary, RDF databases:

- ✓ leave the original data intact
- ✓ increase the speed of data retrieval
- ✓ allow for maximum data flexibility
- ✓ use the international RDF standard developed by the World Wide Web Consortium.

This document was developed with input from:

**Tucana**<sup>™</sup>  
TECHNOLOGIES, INC

11710 Plaza America Drive  
Suite 2000  
Reston VA 20190  
Phone 703-871-5312  
[www.tucanatech.com](http://www.tucanatech.com)

