#2000**
Unity— A Database Integration Tool

Key Features…

- Automatic integration of data sources
- Capture process performed only once per data source using integration software
- Standardized dictionary for use across integration domains
- Structural conflicts handled without user involvement

Summary

TR*Labs* has developed Unity, a software tool for the integration of data sources within a company, across a network, and even on the WWW. Unity automates the integration process by resolving data representation conflicts between systems. Data distribution and organization is hidden from the user.

Background

With the increasing reliance on database systems to store, process, and display data comes the additional problems of using these systems properly. Most organizations have several database systems, which must work together. As data warehouses, data marts, and other Online Analytical Processing (OLAP) systems are added to the mix, the complexity of ensuring interoperability between these systems increases dramatically.

The fundamental problem with distributed data access is the determination of semantically equivalent data. Ideally, users should be able to extract data from multiple sites and have it automatically combined and presented to them in a

usable form. No system has been able to accomplish these goals due to limitations in expressing and capturing data semantics, and industry efforts to standardize communications between systems provide no integration or interoperability between the communicating systems.

The current approach to data source integration is using mediator and wrapper systems, which answer queries across a wide-range of data sources. These systems construct integrated global views, using designer-based approaches, which are mapped using a query language or logical rules into views or queries on the individual data sources. Once an integrated global view and corresponding mappings to source views are logically encoded, wrapper systems are systematically able to query and provide interoperability between diverse data sources.

Unfortunately, mediator and wrapper systems require dedicated database designers and many man-hours of query design and engineering to build a global view for any given multidatabase environment. As a result, database integration is, in many cases, prohibitively expensive and the results are not usually transferable to other multidatabase environments. Further, when data sources are added or removed from the global view, the integration must be re-performed.

Using a standardized dictionary, which provides terms for referencing and categorizing data, TR*Labs* has developed Unity— a software tool for the integration of data sources within a company, across a network, and even on the WWW. Unity automates the integration process by resolving data representation conflicts between systems. Data distribution and organization is hidden from the user.

How the technology works

There are four components of the architecture: a standardized term dictionary, a metadata specification for capturing data semantics, an integration algorithm for combining metadata specifications into an integrated view, and a query processor for resolving structural conflicts at query-time. The dictionary provides a set of terms for describing schema elements and avoiding naming conflicts. The integration algorithm matches concepts to produce an integrated view, and the query processor translates a semantic query on the integrated view to structural query expressions.

Standardized Global Dictionary

To provide a framework for exchanging knowledge, there must be a common language in which to describe the knowledge. Since a computer has no built-in mechanism for associating semantics to words and symbols, an on-line dictionary is required to allow the computer to determine semantically equivalent expressions.

The standard dictionary is organized as a tree of concepts (Figure 1), with all concepts placed into the tree as a generalization/specialization relationship or using a 'part of' relationship (i.e. an address would include a city, province, postal code, etc.) The exact terms and their placement are irrelevant. The dictionary is treated as standard whether within an organization or for the entire Internet community. A database designer associates proper dictionary terms to represent schema element semantics when capturing information about a database to be integrated.

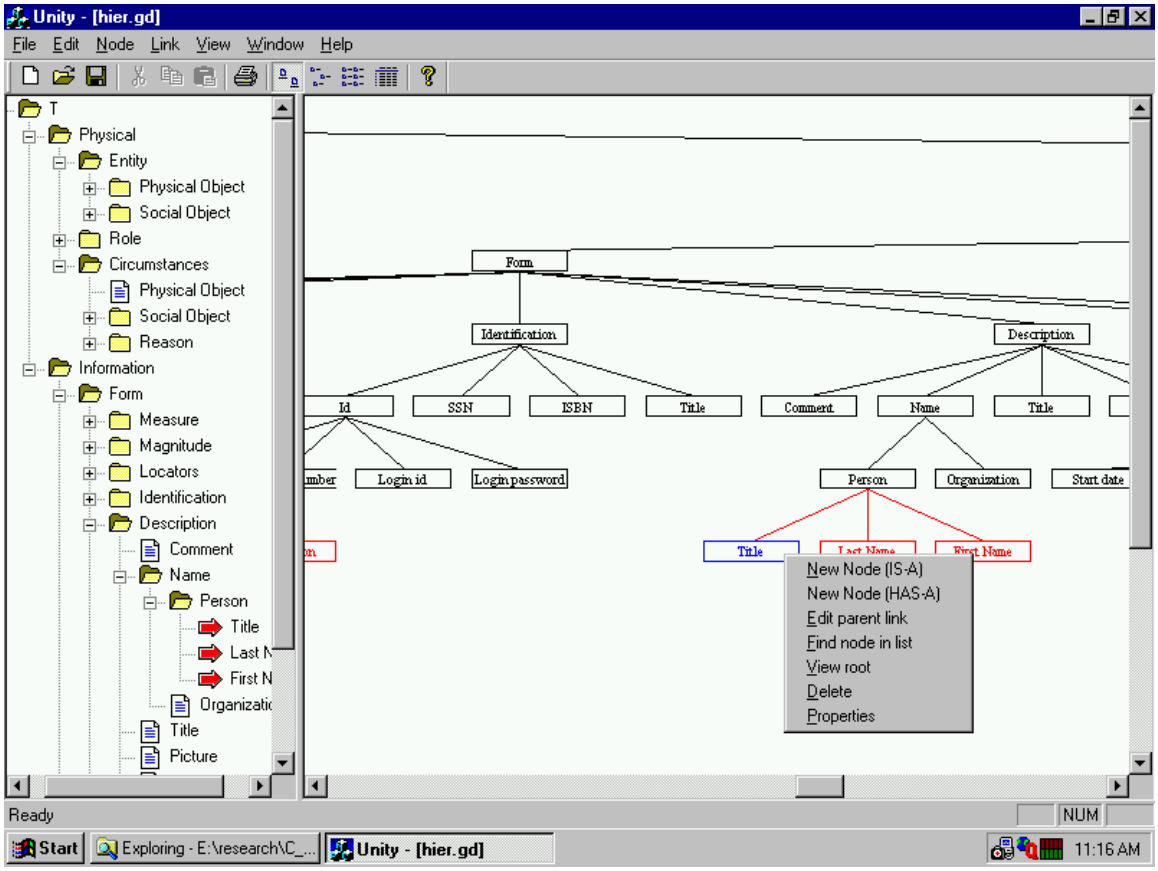

Figure 1 – *Standard Dictionary displayed with Unity*

By analogy, the dictionary is like an English dictionary, as it defines the semantics of accepted words used to convey knowledge. However, overall semantics are communicated by organizing words into a structure such as sentences. The TR*Labs* structure for semantic communication is a semantic name whose simplified structure is easily parsed.

A semantic name captures system-independent semantics of a schema element including contextual information. That is, a semantic name consists of an ordered set of context terms and an optional concept name term. The concept

name is a single, atomic term describing the lowest level semantics. Each context and concept term is a single term from the standardized dictionary.

X-Spec— A Metadata Specification Language

A standard dictionary is not a standard schema as concepts may be represented in different ways in various data sources. Thus, an XML-based specification document called an X-Spec encodes database schema using dictionary terms and additional metadata. X-Specs store relational database schema including keys, relationships, joins, and field semantics.

An X-Spec is constructed using the specification editor component of Unity during a capture process, where the semantics of schema elements are mapped to semantic names. A capture process is performed at each individual database independently of capture processes at other data sources. Thus, a designer does not need to know about the structure of other databases when building the X-Spec for a data source. X-Specs can be constructed in parallel at different sites and at different times.

TR*Labs* has built a specification editor that parses relational schema, formats the information into an X-Spec, and allows the user to include additional information that may not be electronically stored such as relationships and constraints.

XML is used for convenience and interoperability with emerging standards. In summary, an X-Spec is a database schema and metadata encoded in XML that is exchanged between systems and stores semantic names to describe schema elements. Once the X-Specs are constructed all remaining processes are automatic.

Integration Algorithm

The integration algorithm is a straightforward term matching algorithm. The same term in different X-Specs is known to represent the identical concept regardless of its representation. The algorithm receives as input one or more X-Specs and uses the semantic names present to match related concepts.

The integration process is automatic once the capture processes are completed. By their nature, capture processes are partially manual, as they require database designers to capture semantic information in X-Specs. However, once a capture process for a data source is completed, it never has to be re-performed regardless of the other data sources being integrated. This is a significant advantage as it allows database semantics to be captured at design-time. Thus, the advantage of the architecture is that a global view is automatically created once database designers independently define the local views of the individual data sources.

Users access data sources through semantic names, which map to schema elements, and hide the structural representation of concepts in individual databases. The user is thus isolated from the complexities of data distribution, organization, structure, and local naming conventions.

Query Processor

Users generate queries by manipulating semantic names. The user is not responsible for determining schema element mappings, joins between tables in a given data source, or joins across data sources. The system handles the necessary joins based on the relationships between schema elements.

The query processor in Unity determines the semantic names of concepts requested by the user, and for each data source, determines the best field and table mappings for each semantic name. Joins are automatically inserted to connect tables. All the required mapping information is present to construct a select-project-join query, which is translated to SQL and executed on each data source using ODBC. Finally, results retrieved from each data source are formatted for the user and may be joined together based on the presence of common keys.

Project Status

As of November 2000 a working software prototype has been developed. Work is in progress to enhance and upgrade package.

Future Developments

- Possible commercialization of software
- Continued work on expanding scope of automation
- Improved query capabilities and performance

For more Information please contact

Dr. Vinod Ratti,
800 Park Plaza, 10611-98th Ave.,
Edmonton, AB, Canada T5K 2P7

Telephone: (780) 441-3812
Fax: (780) 441-3600
E-mail: vratti@edm.trlabs.ca

Homepage: http://www.trlabs.ca/techshowcase/index.html

Emerging Technology Bulletins are published by TRLabs periodically to inform our sponsors of early stage technologies and invite collaboration towards further development.