



DOME Semantic Integration Suite

Zhan Cui

Copyright © 2002 IBSR BTextact Technologies

BTextact Technologies www.btextact.com Adastral Park, Martlesham, Ipswich, Suffolk IP5 3RE, UK

A trademark of British Telecommunications plc. Registered Office: 81 Newgate Street, London EC1A 7AJ. Registered in England no. 1800000

Scope

This document is intended for decision makers to evaluate the applicability and benefits of DOME for their applications. It includes a description of the problem space and in-depth technical details.

Executive summary

Many business applications built nowadays necessarily require the integration of data from disparate data sources. Although the Internet provides a universal, low-cost infrastructure for sharing data, the time and effort taken to develop new business applications have by no means been reduced. Time is wasted in searching for the right data. Once found, it is often difficult to distil, transform, merge and act upon. The root cause for this is semantic mismatch among disparate data sources. For example, different terms may be used to refer to the same product.

The DOME semantic integration suite has developed an innovative solution to rapidly plug in data sources to form a dynamic content network and to dynamically integrate data to support business applications such as web services that require data from multiple, disparate data sources. Built on available commercial solutions to system, syntactic and structural mismatches, DOME solves the semantic mismatch by accurately representing specific terms semantics of individual data sources, rather than reconciling them to satisfy an integration schema. When data from multiple, disparate data sources need to be integrated, DOME retrieves those semantic definitions, reasons over them, makes deductions where necessary and then produces specific reconciled views for individual applications.

The DOME suite provides a set of graphical user interface (GUI) tools for creating rich, standards-based shared vocabularies, for individual data sources to specialise a vocabulary through constraining term definitions or introducing new terms, and for creating mappings between vocabularies. The DOME suite is designed to allow disparate data sources to be integrated in a way that: -

- Data sources can join or leave the server freely without re-engineering the server or other on-line data sources,
- existing applications can use data from newly joined data sources without re-engineering,
- concept-based queries can be answered,
- semantic join is possible among independent data sources,
- mismatches among data sources are resolved in context,
- DOME services can be accessed over the web,
- open, industrial-strength, standards are used wherever possible.

The DOME suite is a valuable resource for any enterprise that needs to manage heterogeneous information systems to share data or knowledge. It can support intra/inter-organisational business services, help build new applications such as data warehouses and support dynamic e-catalogue integration, to name but a few of its many application areas.

Key value-adds of DOME are: -

- The flexibility to accurately describe data source contents, in particular to differentiate products,
- provision of different level of abstractions of the underlying data sources,
- reuse and sharing of declarative mapping rules,
- support for ad-hoc queries,
- streamlining of data source joining processes.

Glossaries

API	Application Programming Interface
CORBA	Common Object Request Broker Architecture
DCOM	Distributed Common Object Model
DOM	Domain Ontology Management Environment
DTD	Document Type Definition
EJB	Enterprise Java Bean
J2EE	Java TM 2 Platform, Enterprise Edition
JDBC	Java Data Base Connectivity
ODBC	Open Data Base Connectivity
SOAP	Simple Object Access Protocol
UDDI	Universal Description, Discovery and Integration
UNSPSC	United Nation Standard Product and Service Classification
XML	eXtensible Markup Language

Information sharing today - Bridging the heterogeneity of information sources

Many business applications built nowadays necessarily require the integration of data from disparate data sources. Although the Internet provides a universal, low cost infrastructure for data sharing, the time and effort taken to develop new business applications have by no means been reduced. The cost of information reuse is high: time is wasted in searching for the right data; once found, it is often difficult to distil, transform, merge and act upon.

The main cause of these problems is heterogeneity, which can be classified roughly as system, syntactic, structural and semantic heterogeneity.

- System heterogeneity includes incompatible hardware and operating systems;
- syntactic heterogeneity refers to different languages and data representations;
- structural heterogeneity includes different data models such as relational and object-oriented models;
- semantic heterogeneity refers to the meaning of terms.

Examples of semantic heterogeneity are that the same product may have different names within different communities, and different products may have the same names. Locally, abbreviations may be more convenient and appropriate. There are also mismatches between terms used by humans and machines. For example, a single conceptual schema is often implemented using several database schemas. Importantly, many data are stored under assumptions that are either separately documented or not documented at all; thus data cannot easily be reused out of their boundaries, i.e., contexts.

These problems are increasing as companies go global through mergers and acquisitions, necessitating the integration of different enterprise information systems. Similarly, electronic commerce and the automated linking of supply chains require seamless access to information across disparate data sources.

Considerable progress has been made in addressing the problems associated with the use of different hardware, software, syntax and even data models. Middleware and related standards such as CORBA, DCOM, ODBC, JDBC and J2EE work well, and have already been widely deployed. Recently, XML/DTD and related standards such as SOAP and UDDI have also gained popularity, and these provide de facto syntax, structure and protocols for representing, transferring and retrieving contents and services.

By building on these technologies, we have developed the Domain Ontology Management Environment (DOME): an innovative solution to deal with the semantic heterogeneity that is considered to be a major roadblock to open electronic commerce and information sharing. Most current methods such as data warehouses, federated databases and XML require common standards or integrated schemas to achieve semantic integration. The mismatches among data sources are manually engineered out and mapped to integrated schemas through custom-developed programs. This works - and works well - when the applications and data sources are known, or could be anticipated at design time.

The problems associated with this approach, among others, are: -

- Long development cycles, thus costly in terms of time and resources. New applications often require the building of new data warehouses and rework on each data source;
- commoditisation of information or contents, thus in particular limiting a supplier's ability to differentiate its products from those of its rivals;

- design-time semantic mismatch reconciliation is too limiting because many mismatches have to be resolved in context. For example, if both buyers and suppliers are based in UK, there is no need to perform currency conversion. If a UK buyer searches for products from international suppliers, currency conversion may be required;
- standards are unlikely to meet dynamic requirements of eMarketplaces due to the slow consensus making process involved.

The deficiency of these techniques to handle semantic heterogeneity has prompted users to build their own solutions. Many local knowledge bases are created to capture the semantics of data sources and their mappings to the integrated schemas, in the hope of facilitating their reuse in other projects. Clearly, there is an opportunity to: -

- Capture these semantics in an integrated environment,
- Use tools to streamline and simplify the process of adding independent data sources to form a pool of shared information,
- Avoid costly system re-engineering or establishment of new databases,
- Dynamically gather data and reconcile mismatches in context.

The DOME approach - Representing the semantic heterogeneity

DOME's solution to the semantic challenge is the DOME semantic integration suite, shown in Figure 1. The suite provides customisable, unified, conceptual and consistent views of the underlying information systems for both users and applications. It allows user GUIs and applications to access the DOME semantic integration services through the same set of APIs. One of the DOME services is to process concept-based queries to the underlying and distributed information systems without the users knowing the heterogeneity among them.

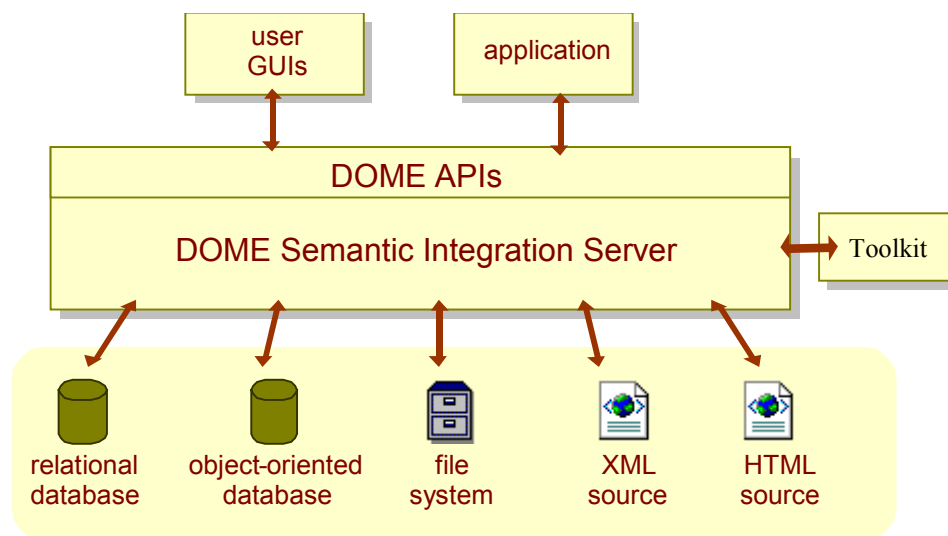


Figure 1: Basic DOME semantic integration suite

Built on available commercial solutions to system, syntactic, structural and data model heterogeneity, DOME solves semantic heterogeneity by representing specific semantics of terms used by individual information systems, rather than reconciling them to satisfy an integration schema. DOME offers not only a set of vocabularies rich enough to model many different data sources, but also the ability to allow individual data sources to specialise a vocabulary through constraining term semantics or introducing new terms.

The DOME suite supports the creation of rich, standards-based shared vocabularies that can be used to model various information systems and applications. It empowers developers to choose appropriate terms from their shared vocabulary to describe data sources accurately, and without the need to either compromise or worry about how to reconcile the mismatches with other data sources. The DOME suite provides a unique solution for resolving semantic mismatches by reasoning over disparate data source descriptions and producing specific reconciled views for individual applications.

The DOME suite has been developed to tackle problems arising from semantic heterogeneity among disparate and independently developed information systems. It also addresses practical issues to satisfy the requirements of real world B2B electronic commerce applications.

The semantics of any standards and vocabulary changes over time. The DOME suite provides easy to use GUI tools to maintain and update shared vocabularies as well as to customise them to suit local usage.

The DOME suite streamlines and simplifies the process of adding independent data sources to the DOME semantic integration server. When adding a data source, the administrator of the data source first selects a shared vocabulary where necessary, to customise it to produce a localised vocabulary, which could be used to describe the data source. This localised vocabulary defines the domain-specific interpretation of terms that will be used by DOME when it is integrated with other data sources. This is different from the current approach, which requires the data source to transform their data to an integrated schema. This also enables DOME to perform context-based mismatch reconciliation at run time.

DOME recognises the current limitations of semantic or meaning definitions. DOME uses mappings to make the linkage when recorded semantics are not enough to map from one term to another. As mapping definition is a major engineering effort, DOME provides mapping suggestions and easy-to-use GUIs to allow administrators to create mappings, update mappings visually as well as with the ability to type in pre and post-conditions. DOME also check the correctness of these mappings and make suggestions whenever possible.

DOME allows the users, as well as applications, to use the vocabularies customised from shared vocabularies to issue ad hoc concept-based queries which previously are time-consuming or cost prohibitive to do using other systems. DOME then queries the distributed data sources on behalf of the users and applications. DOME performs necessary operations to join data and aggregate them across different data sources. The DOME semantic integration server sometimes has to seek a third data source and make deductions in order to join two independent data sources.

DOME key technology description

The key technology that DOME uses to define vocabulary is ontology. An ontology is a formal specification of a conceptualisation. A conceptualisation includes real or abstraction entities and the relationships between them. The specification includes terms used to denote the entities, relationships between entities such as term inheritance, formal structures of entities such as attributes of an entity, and any constraints. The use of ontology enables DOME to check the consistency of shared vocabularies, make deductions on the relationships between vocabularies and resolve mismatches in context.

DOME uses the following ontologies to meet information sharing requirements: -

- *Shared ontologies* - are used to define shared vocabularies.
- *Source ontologies* - are data source-specific vocabularies that contain source-specific interpretations of the terms of shared vocabularies.
- *User and application ontologies* - they are the same as source ontologies except that they are attached to users and applications respectively.

The inter-relationships between the various ontologies are shown in Figure 2. The single arrows denote inheritance and double arrows denote information flow.

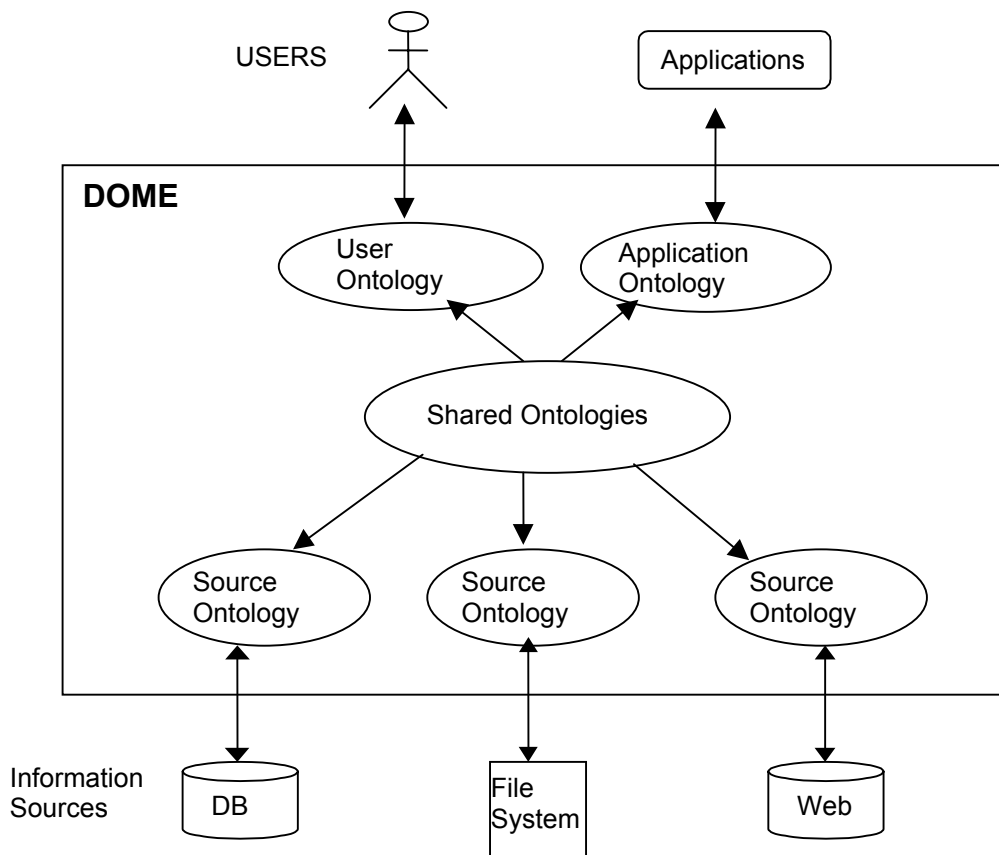


Figure 2: Models of data sources and their relationship

Shared ontologies

A shared ontology is a common, agreed vocabulary of both the domain users and developers. Shared ontologies can be developed from open standards (e.g. UNSPSC), enterprise information models or in-house 'standards'. The DOME suite maintains a set of shared ontologies.

Shared ontologies enable DOME to provide unified and consistent views of the underlying distributed and heterogeneous data sources, to accept and process ad hoc and concept-based queries, to make deductions about data, to transform and merge data and to resolve semantic mismatches together with source, user and application ontologies.

Source, user and application ontologies

Source ontologies define the data semantics of their associated data sources. The terms in source ontologies are often taken from a shared ontology, but their definitions could be further constrained. For example, certain attributes may be sub-typed or have fixed values. New terms may be defined, over and above those in the shared ontology. In DOME a data source could be associated with only one source ontology. One source ontology may be associated with more than one data source.

The following example shows the relationships between those ontologies. Let's say the shared ontology defines a concept *Price* as follows: -

- Price:*
- *Amount: real*
 - *Currency-type: currency-type*
 - *Scale-factor: real*

Let's also assume all the concepts used in defining *Price* are also defined in the shared ontology and they are defined as primitives. The semantics of primitives are not defined. They require human-level agreements.

The concept *Price* could be instantiated in the following ways: -

Source ontology 1:

- Price:*
- *Amount: real*
 - *Currency-type: US-Dollar*
 - *Scale-factor: 1*

Source ontology 2:

- Price:*
- *Amount: integer*
 - *Currency-type: Yen*
 - *Scale-factor: 1000*

User ontology 3:

- Price:*
- *Amount: real*
 - *Currency-type: GBP*
 - *Scale-factor: 1*

Shared ontologies, and source, user and application ontologies are the critical components for DOME to perform context-based mismatch reconciliation at run-time. Specifically, mismatches are resolved according to the semantic definitions of the ontology used to construct queries.

More specifically, let's use the currency as an example. For a UK buyer, when all suppliers are using GBP as their currency, there will not be any currency type conversion. If there is an international supplier using, say, Yen, these prices will be converted to GBP for the UK buyer without the buyer knowing it. Of course, the buyer could force the system not to resolve any mismatch, if they prefer. Similarly scale factors are also mediated.

DOMÉ suite description

The DOME suite, as shown in Figure 3, consists of a semantic integration server, client software (which leverages the DOME APIs) and wrappers for data sources of various types. The semantic integration server includes the Middleware necessary to support the integration of heterogeneous data sources. There are separate client tools for setting up a network of integrated data sources and for issuing queries over the set of the data sources. External application programs can access the services provided by semantic integration server via the DOME APIs.

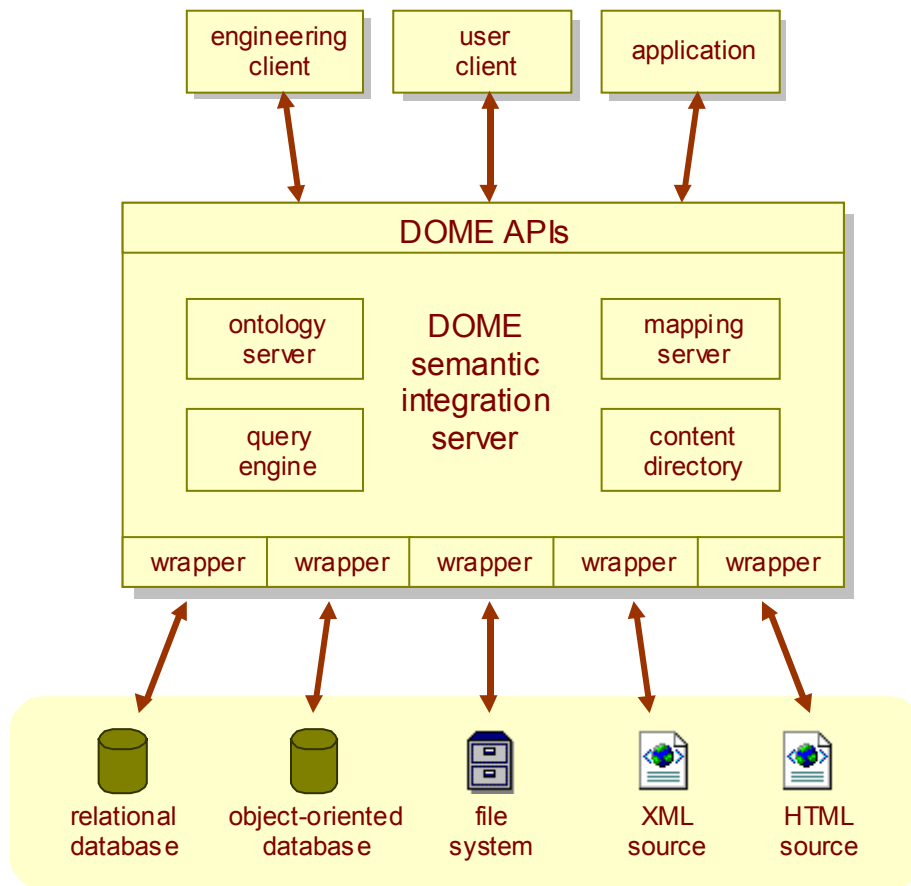


Figure 3: Architecture and principle components of DOME suite

Semantic Integration Server

The semantic integration server includes a query engine that solves queries by retrieving and joining data from different sources, a directory that maintains up-to-date records of currently available data, and wrapper technology to allow communication with a variety of databases or web-based resources. It also manages various vocabularies (known as ‘ontologies’); user vocabularies, which allow users to ask queries in their own terminology; and a shared vocabulary, which acts as a ‘lingua franca’ between users and data sources. A mapping server manages the mappings between different vocabularies.

The semantic integration server is implemented as an Enterprise JavaBean™ and can be deployed on any J2EE™-compatible EJB™ container.

Engineering client

The suite’s engineering client acts as a ‘wizard’ that streamlines the process of adding data sources to the DOME server. It features tools for creating shared vocabularies, personalising a shared vocabulary, specifying mappings, and configuring wrappers. It is implemented using JavaServer Pages™ and Java™ Servlets.

User client

The user client allows users to query the integrated information. It has a graphical user interface for browsing common vocabulary of the integrated data sources, and allows the user to ask concept-based queries, hiding the details of the underlying sources from users. It is implemented using JavaServer Pages™ and Java™ Servlets.

Wrappers

The wrappers perform translations of queries expressed in the DOME query syntax and terminology of the resource ontology to queries expressed in the syntax of the resource query language and the terminology of the resource schema. Although they are configured for particular resources, DOME wrappers are generic across resources of the same type; for example, Wrappers of SQL database utilise the same code.

Ontology server

The ontology server stores, and allows access to, all the different kinds of ontologies that are defined using the engineering client (described above): namely *shared*, *source*, *application* and *user*. Its ontology engine checks the ontology consistency when term definitions are modified or new terms are added.

Mapping server

The mapping server stores any mappings between ontologies that are defined by the engineer in setting up a DOME network. The mapping server also stores generic conversion functions that can be utilised by the engineer when defining a mapping from one ontology to another. These mappings are specified using a declarative syntax, which allows them to be straightforwardly modified and reused. The query engine queries the mapping server when it needs to translate between ontologies to resolve a query.

Content directory

When a source is connected to a DOME network, its wrapper will inform the DOME content directory about its existence and pass to it a description of the contents of the source, expressed in terms of the relevant source ontology. This ensures that the query engine is able to identify what information is available without having to access the schema of the source. When a wrapper is - for whatever reason - no longer able to provide information from a source, it will inform the content directory, which is then able to discount that source from any future query solving.

Query engine

Upon receiving a query, the DOME query engine first needs to decide which sources are relevant to that query. It obtains a list of currently available and relevant sources by consulting the content directory. Based on this information, the query engine does the following:

- It decomposes the query into sub-queries in such a way that the results of the sub-queries can be integrated;
- It translates each sub-query from the ontology of the original query to that of the relevant source;
- It sends each sub-queries to the relevant data source;
- It translates and then integrates the results of the sub-queries.

Thus the user or application making the query receives the results expressed in the same terminology that the query was expressed in, and is oblivious of the details of the underlying data sources.

Ontology extractor

DOME also includes an ontology extractor that is not shown in Figure 3. The ontology extractor is built on the software re-engineering and artificial intelligence (AI) technologies. When given database schemas and a shared ontology it will automatically produce an ontology that could be used, after human checking and modification, as a source ontology for the given database. The ontology extractor could also extract information from application

programs to enrich the source ontology. Recently the ontology extractor has been extended to extract information from web resources.

Summary and DOME value-adds

DOME is a semantic integration server designed to allow heterogeneous data sources to be integrated in a way that:

- Data sources can join or leave the server freely without re-engineering the server or other on-line data sources,
- existing applications can use data from newly joined data sources without re-engineering,
- concept-based queries can be answered,
- semantic join is possible among independent data sources,
- mismatches among data sources are resolved in context,
- DOME services can be accessed over the web,
- open, industrial-strength, standards are used wherever possible.

The DOME suite is a valuable resource for any enterprise that needs to manage heterogeneous information systems to share data or knowledge. It can support intra/inter-organisational business services, help build new applications such as data warehouses, and support dynamic e-catalogue integration, to name but a few of its many application areas.

Key value-adds of DOME are:

- The flexibility to accurately describe data source contents, in particular to differentiate products,
- provision of different level of abstractions of the underlying data sources,
- reuse and sharing of declarative mapping rules,
- support for ad-hoc queries,
- streamlining of data source joining processes.

BTexact Technologies helps businesses and organisations gain maximum advantage from communications technology. We create value and competitive advantage by combining a deep knowledge of networks and networked applications with proven skills in business consulting, change management and innovation.

BTexact Technologies www.btexact.com Adastral Park, Martlesham, Ipswich, Suffolk IP5 3RE, UK
Email: btexact@bt.com Freephone: 0800 169 1689 (UK only) Phone: +44(0) 1473 607080 Fax: +44(0) 1473 607700

BTexact Technologies is a trademark of British Telecommunications plc. Registered office: 81 Newgate Street, London EC1A 7AJ. Registered in England No: 1800000. The products and services described in this publication are subject to availability and may be modified from time to time. Products and services are provided subject to British Telecommunications plc's respective standard conditions of contracts. Nothing in this publication forms any part of any contract. All third-party trademarks are hereby acknowledged. © British Telecommunications plc, 2002