Interoperability of Legacy Databases
A Combined Top-Down and Bottom-Up Approach

Ph. Thiran (PhD Student) and J-L. Hainaut (PhD Supervisor)
InterDB Project[1], Database Applications Engineering Laboratory
Computer Science Department, University of Namur, Belgium
{pth, jlh}@info.fundp.ac.be

Abstract

The thesis focuses on the interoperability of autonomous legacy databases with the idea of meeting the actual requirements of an organization. The interoperability is resolved by combining the top-down and bottom-up strategies. The legacy objects are extracted from the existing databases through a database reverse engineering process. The business objects are defined by both the organization requirements and the integration of the legacy objects.

1. Introduction

Most large organizations maintain their data in many distinct independent databases that have been developed at different times on different platforms and DMS (Data Management Systems).The new economic challenges force enterprises to integrate their functions and therefore their information systems including databases they are based on. In most cases, these databases cannot be replaced with a unique system, nor even reengineered due to the high financial and organizational costs of such a restructuring.

We refer to software services allowing such so-called **legacy database systems** to cooperate, as providing **interoperability**. Such services provide users and application programs with an integrated view of data dispersed over various component databases.

In this thesis, we focus on the interoperability of **autonomous legacy databases** with the idea of meeting the actual requirements of an organization. We introduce the thesis by first giving the main issues about interoperability. Next, a short overview of the interoperability research is presented. Finally, we present the purpose, the topic and the state of the thesis.

[1] The InterDB Project is supported by the Belgian *Région Wallonne*.

2. Problems and context of the interoperability

2.1 Legacy database systems

The presence of legacy data systems is one of the major obstacles in the use of integrated information [Brodie, 1995]. Typically, legacy data systems are very large. They are typically written in old programming language like COBOL or PL/1. Such systems are usually mission critical and inflexible in nature [Bouguettaya,1998].

Integrating such systems is very costly because of the complexity of understanding data semantics which is either buried in application programs or was never documented by original designer. The incompleteness of their specifications leads to ambiguities of the interpretation of the data schema. The hardest case is when data resides in files, but understanding unmormalized and poorly documented relational databases also is very difficult ([Hainaut, 1996], [Parent, 1998]).

2.2 Autonomy

Legacy database systems were typically designed to support local requirements imposed by the local environment, and without considering a possible cooperation with other systems. In other words, databases are usually under separate and independent control. The different aspects of autonomy are summarized as follows [Sheth, 1990]:

1. **Design autonomy.** The databases have their own data model, query language, semantic interpretation of data, constraints, etc.
2. **Communication autonomy.** The databases have the ability to decide when and how to respond to requests from other databases.
3. **Execution autonomy.** The execution order of transaction is controlled by the legacy databases. They don't need to inform any other system of the execution order of local or external operations.
4. **Association autonomy.** The legacy databases are able to decide whether participate or not in one or more federations, as well the possibility of its dissociation of a federation.

2.3 Heterogeneity

A major obstacle to interoperability of legacy databases is their **heterogeneity**. Heterogeneity among legacy databases is caused by the design autonomy of their owners in developing such systems. Legacy systems were typically designed to support local requirements, under constraints imposed with a given system.

We can distinguish several types of heterogeneity [Thiran, 1998]: platform, DMS, location and semantics level. The **platform** level copes with the fact that databases reside on different brands of hardware, under different operating systems, and interacting through various network protocols. Leveling these differences leads to platform independence. **DMS level** independence allows programmers to ignore the technical detail of data implementation in a definite family of models or among different data models. **Location** independence isolates the user from knowing where the data reside. Finally, **semantic** level solves the problem of multiple, replicated and conflicting representations of similar facts.

Current technologies such as de facto standards (e.g. ODBC and JDBC), or formal bodies proposals (e.g. CORBA, EJB), now ensure a high level of platform independence at a reasonable cost, so that this level can be ignored from now on. DMS level independence is effective for some families of DBMS (e.g. through ODBC or JDBC for RDB), but the general problem is still unsolved when several DMS models, including legacy ones, are to cooperate. Location independence is addressed either by specific DBMS (e.g. distributed RDBMS) or through distributed object managers such as CORBA middleware products. Despite much effort spent by the scientific community, semantic independence still is an open and largely unsolved problem ([Aslan, 1999], [Härder, 1999]).

3. Interoperability and mediation

3.1 Mediation

To address the problem of interoperability of information systems in general, the term **mediation** has been defined in [Wiederhold, 1995] as a service that links data resources and application programs. A **mediator** is a software module that exploits encoded knowledge about some sets or subsets of data to create information for applications [Wiederhold, 1992]. Tasks involved in mediation include [Vermeer, 1996]: (1) accessing and retrieving relevant data from multiple heterogeneous sources, (2) transforming retrieved data to be integrated, (3) integrated the homogenized data, (4) managing the instance and structural conflicts, and (5) reducing the integrated data by abstraction. Several prototype mediator systems have been developed (e.g., [Garcia, 1995], [Vermeer, 1996]).

3.2 Mediation and legacy databases

A legacy database federation can be seen as a special case of mediation, where all data sources are legacy databases (i.e., heterogeneous and autonomous) and the mediator offers a virtual and integrated view of the underlying legacy databases.
A legacy database federation performs mediation by using a hierarchy of mediators that dynamically transform queries based on a federated schema into physical queries based on the physical schema of the legacy database sources (Cf. Figure 1).

3.2.1 Hierarchy architecture

The hierarchy architecture of a federation in general has been described in [Sheth, 1990]. It consists of a hierarchy of data descriptions that ensure independence according to different dimensions of heterogeneity. According to this framework and according to the legacy nature of the database source, each local database source is described by its own **physical schema** from which a semantically rich description called **conceptual schema**, is obtained through a database reverse-engineering process. From this conceptual view, a subset called **export schema** is extracted. All the export schemas are merged into **the federated schema**. The federated schema as well as the conceptual and export schemas are expressed in a **canonical data model** which is independent of the underlying technologies.

3.2.2 Component architecture

The function of a **mediator** is to provide integrated information, without the need to integrate the data resources. A mediator hides details about the location and representation of relevant data to applications.

On top of each legacy database is a **wrapper**. A wrapper is a software component that performs the translation between the export schema and the physical schema of the database [Papakonstantinou, 1995]. That is, the wrapper (1) offers an export schema in the canonical data model (2) accepts queries against the export schema and translates them into queries understandable by the underlying database, and (3) transforms the results of the local queries into a format understood by the application. Wrappers and mediators relies on schema descriptions and mappings to translate queries and to form the result instances.

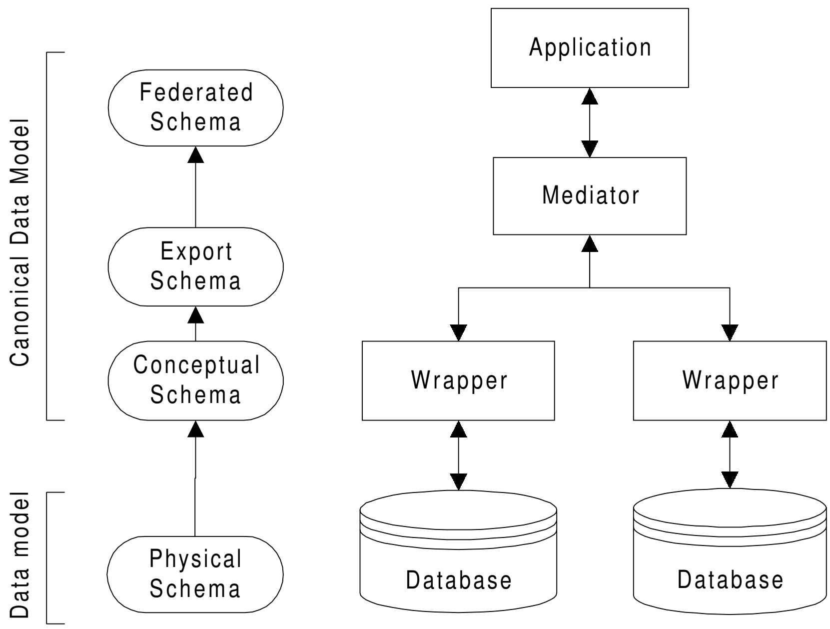

Figure 1 – A general architecture of a database federation

3.2.3 Heterogeneity issues

The architecture model depicted in Figure 1 provides an adequate framework for solving the heterogeneity issues discussed above [Thiran, 1998]. DMS and local semantic independence is guaranteed by the wrappers. Location and global semantic independence is ensured by the mediators. It provides data federated access irrespective of their location and resolves semantic conflicts. Finally, platform independence is ensured by both the wrappers and ad hoc middleware such as commercial ORB.

3.2.4 Object-oriented database federation

Today, the use of object-oriented techniques in building database federations has been widely accepted. Many approaches use such techniques in their system architecture. Moreover, object-oriented data models are commonly used as the canonical data model for database federations. However, Object-oriented models do not provide rich data structuring possibilities which enables them to express all the semantics of a local schema expressed in other data models [Conrad, 1999]. Hence, the need for richer model.

3.3 Mediation and methodology

The current methodologies developed for building a database federation are generally based on a database integration approach (e.g., [Sheth, 1990], [Schmitt, 1996], [Parent, 1998], [Hainaut, 1999]). It produces the structure of the federated schema that depends directly on the integrated export schemas and the integration method used [Busse, 2000].

As discussed in [Hasselbring, 1999], this bottom-up approach exhibits the following problems:

- The process of integration is often more complex than required for the actual requirements of the organizations. Since the relevant information is hidden in a federated schema, the user is responsible for finding the required information.
- It rarely considers the requirements from the new applications that are to be developed on top of the legacy databases.
- It is not suitable for frequent dynamic changes of organization requirements since the federated schema is static or usually too difficult to change.

4. Purpose and scope of this thesis

We mainly focus on the **semantic aspects of interoperability of legacy databases**. We usually abstract from wrapper-mediator architecture described in [Wiederhold, 1995]. We do not assume, however, the database federation as the result of a bottom-up process. Referring to [van den Heuvel, 2000] and [Busse, 2000], we propose to resolve the interoperability by combining the top-down and bottom-up processes. The export schemas are still obtained during a bottom-up process but the federated schemas are determined through a top-down process. The integration process is therefore limited to a linking mechanism.

In summary, we address the following three essential topics for building a database federation:

1. Since we take into account the actual requirements of an organization, we integrate the notion of **business objects** in the federation. Therefore, we consider the federated schema as the result of a **forward engineering process** that captures the semantics of business, in a way that is very close to the business reality.
2. We study also the extraction of **legacy objects** from existing databases through the **database reverse engineering process**. Our goal is to exploit a semantically rich description of the federation components in order to more properly detect correspondences between the export schemas ([Ramesh, 1995], [Thiran, 1998], [Conrad, 1999]).
3. Motivated by both the top-down and bottom-up processes, we investigate the **correspondence** assertions which explicitly specify the relationships between the business and legacy objects.

Our primary goal of this thesis is to introduce the concept of business object in the legacy database federation. Hence, we state that a business object is defined by both the organization requirements and the description of the federation databases.

One of the most challenging issues is the definition of the mappings between all the schemas of a federation. In particular, we linger over the linking of business and legacy objects. In this thesis, we develop a formal **transformational approach** that is built on an unique extended object-oriented model from which several abstract submodels can be derived by specialization. This approach is intended to provide an elegant way to unify the multiple models and mapping descriptions of the federation. Our goal is then to elaborate techniques and reasoning common to database federations, trying to provide a **general approach** for developing federations.

In the following, we briefly summarize the basic assumptions and restrictions underlying this thesis:

1. First of all, we do not address issues such as infrastructure and transaction. For such issues, we refer to [Sheck, 1991] and [Deacon, 1996].
2. Moreover, we concentrate on the structural part of the legacy schemas to be integrated. Dynamic constraints which describe restrictions on the legacy database behavior are beyond the scope of this thesis. For an

overview and discussion on behavior integration, we refer to [Vermeer, 1996].

3. Furthermore, we consider the three-layer architecture for business objects components with the layers: presentation, business process and business entity [Casanave, 1996]. In this thesis, we consider only the business entity object and assume that it is the correct translation of the organization requirements. For the problem of design of business objects, we refer to the existing literature, for instance [Sutherland, 1997] and [Eeles, 1998].

5. Thesis overview

5.1 Thesis topic

We can now define the topic of this thesis as follows:

> How can we provide interoperability for **legacy databases** in the presence of **heterogeneity**, using a combined **bottom-up and top-down methodology** ?

We distinguish three main tasks addressing this question: (1) defining a meta-model intended to express all the federation schemas and mappings; (2) defining a generic architecture of business objects federation; (3) proposing a methodology based on a bottom-up and top-down approaches.

5.2 Defining a meta-model

This issue is discussed in Part I of this thesis. It involves defining a unique and generic meta-model intended to express: (1) the federated model as business objects; (2) the export and conceptual models as legacy objects; (3) the different physical models; and (4) the mappings.

To formally define the mappings between these models, we adopt and develop the schema transformational approach [Hainaut, 1996]. This is based on the work first presented in [Thiran, 1998].

This meta-model is an **abstract formalism** from which the federation models can be derived by specialization. In short, physical schemas, conceptual schemas, export schemas as well as federated schemas are expressed into an unique and generic entity/object-relationship model. Besides the standard concepts, the meta-model includes some **meta-objects** which can be customized according to specific needs. These features provide dynamic extensibility of the generic model. For instance, new concepts such as correspondence types can be represented by specializing the meta-objects.

5.3 Defining an open architecture

This issue is discussed in Part II of this thesis. We assume a wrapper/mediator architecture and extend it to take into account the business object concept. To this end, we provide an overview of existing works on wrapper/mediator and business object. We refer particularly to the general federation architecture that we have developed in [Hainaut, 1999].

We discuss the important role of the wrapper in the particular case of a legacy database federation. This is based on the work presented in [Thiran, 1999]. We discuss also how the concept of business object can be integrated in a wrapper/mediator architecture. That is, we argue that it would be useful for a mediator to be open to the requirements of the organizations instead of being built only from the integration of legacy databases.

5.4 Proposing a general methodology

This issue is discussed in Part III of the thesis. We propose a combined top-down and bottom-up strategy. Referring to the architecture of Figure 1, we distinguish several main tasks for building the architecture presented in the previous section: (1) defining the federated schemas as business objects through forward engineering; (2) recovering the conceptual schemas of the legacy databases and extracting their export schemas through reverse engineering; (3) developing the correspondences between the federated schemas and export schemas; and (4) defining the mappings.

5.4.1 Forward engineering

As previously suggested in the Section 4, we do not focus on the design of the business objects. We focus on **entity business object** definition. An important characteristic of such objects is the explicit separation of interface and implementation [Eeles, 1998]. Hence we reduce the design of an entity business object to the **definition of its interface**.

5.4.2 Reverse engineering

Since we assume that the databases are **legacy systems**, we focus on the semantics recovery Extracting a semantically rich description from a data source is the main goal of the **data-centered reverse engineering process** (DBRE). A general DBRE methodology has been developed in our laboratory (see e.g. [Hainaut, 1996b], [Hainaut, 1999]). During this process, physical schemas that correspond to the legacy databases are translated into conceptual schemas using the entity-object relationship model. However, we don't assume the quality and the completeness of the physical schemas. We get down to detect undeclared constructs and constraints in order to provide a semantically rich description of the legacy databases, and hence to more properly detect correspondences between the conceptual schemas. We also define the **export schemas** that hold the information relevant only to the federation. This is based on the work presented in [Thiran, 1998] and [Thiran, 1999].

5.4.3 Defining the correspondences

This is the process of identifying the objects in different export schemas which are related to federated schemas. To this end, we show the important role of the vertical and horizontal correspondences to define the relationships between export and business objects (Cf. Figure 2). **Horizontal correspondences** are a result of the reverse-engineering. They state the relationships between the export schemas and fall into three possible categories: syntactic, semantic and instance [Thiran, 1998]. On the other hand, **vertical correspondences** explicitly specify the different relationships between the export and federated schemas.

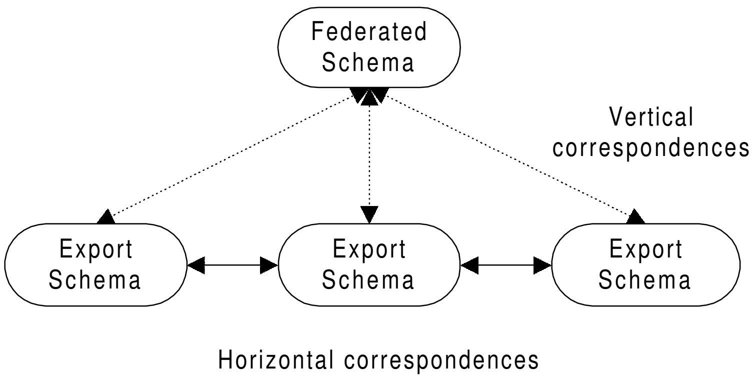

Figure 2 - Horizontal and vertical correspondences

5.4.4 Defining the mappings

Referring to [Hainaut, 1996] and [Thiran, 1998], we argue that it would be possible to define the mappings from **schema transformation**. We assume that deriving a schema to another is performed through techniques such as renaming, translating, solving conflicts which basically are schema transformations. We assume also that defining the correspondences between the export and federated schemas can be formalized as a chain of transformations. To this end, an inventory of useful transformations is presented.
Note that the mappings are defined as transformational functions; they cannot be immediately translated into executable procedures. However, [Hainaut, 1996] shows that it is fairly easy to produce procedural data conversion programs.

6. State of this thesis

This thesis is based on the works that have been developed as part of the InterDB project [Thiran, 1998]. In [Thiran, 1998] and [Hainaut, 1999], we propose the baselines for an architecture, a methodology, including schema recovery through reverse engineering, database integration and mapping building, for the development of a legacy database federation. The methodology is based on a generic model of data schemas and a formal transformational approach to schema engineering. This approach formally defines the mappings between the federation schemas, so that, it is possible to derive the wrappers [Thiran, 1999] and the mediators from them in a systematic way. Moreover, the methodology is supported by the DB-MAIN CASE tool [Hick, 1999] that helps developers generate the wrappers and the mediators.

Acknowledgement

We thank Vincent Englebert for his fruitful discussion and helpful comments.

References

[Aslan, 1999] G. Aslan, D. McLeod, *"Semantic Heterogeneity Resolution in Federated Databases by Metadata Implementaion and stepwise evolution"*, The VLDB Journal, Vol. 8, pp. 120-132, 1999.

[Bouguettaya, 1998] A. Bouguettaya, B. Benetallah, A. Elmagarmid, *"Interconnecting Heterogeneous Information Systems"*, Kluwer Academic Publishers, 1998.

[Brodie, 1995] M. Brodie, M. Stonebraker, *"Migrating Legacy Systems"*, Morgan Kaufmann, 1995

[Busse, 2000] S. Busse, R-D. Kutsche, U. Leser, "Strategies for the Conceptual Design of Federated Information Systems", in *Proceedings of EFIS'00*, pp. 23-32, IOS Press and Infix, 2000.

[Casanave, 1996] C. Casanave, *"OMG Common Business Objects and Business Object Facility RFP"*, OMG Document CF-91-01-04, 1996.

[Conrad, 1999] S. Conrad, W. Hasselbring, U. Hohenstein, R-D. Kutsche, M. Roantree, G. Saake, F. Saltor, “Engineering Federated Information Systems – Report of EFIS’99 Workshop”, *ACM SIGMOD Record*, 28(3), 1999.

[Deacon, 1996] A. Deacon, H-J. Sheck, G. Weikum, "Semantics-based Multi-level Transaction Management in Federated Systems" in *Proc. of 9th Conference on Parallel and Distributed Computing Systems*, pp. 759-765, Releigh, 1996.

[Eeles, 1998] P. Eeles and O. Sims, *"Building Business Objects"*, John Wiley & Sons, New-York, 1998.

[Garcia, 1997] H. Garcia-Molina , Y. Papakonstantinou , D. Quass , A. Rajaraman , Y. Sagiv, J. Ullman, V. Vassalos , J. Widom, “The TSIMMIS approach to mediation: Data models and Languages”, *Journal of Intelligent Information Systems*, 1997.

[Sheth, 1990] A.P. Sheth and J.A. Larson "Federated Database Systems for Managing Distributed, Heterogeneous and Autonomous Databases", *ACM Computing Surveys*, 22(3):183-236, September 1990.

[Hainaut, 1996] J-L. Hainaut, "Specification preservation in schema transformations - Application to semantics and statistics", *Data & Knowledge Engineering*, Elsevier Science Publish, 16(1), 1996.

[Hainaut, 1996b] J-L. Hainaut, J. Henrard, J-M. Hick, D. Roland, V. Englebert, Database Design Recovery, in *Proc. of the 8th Conf. on Advanced Information Systems Engineering* (CAiSE'96), Springer-Verlag, 1996.

[Hainaut, 1999] J-L. Hainaut, Ph. Thiran, J-M. Hick, S. Bodart, A. Deflorenne, "Methodology and CASE tools for the development of federated databases", the *International Journal of Cooperative Information Systems*, Volume 8(2-3), pp. 169-194, World Scientific, June and September, 1999.

[Härder, 1999] Th. Härder, G. Sauter, J. Thomas, *"The Intrinsic Problems of Structural Heterogeneity and an Approach to their Solution"*, The VLDB Journal, Vol. 8, pp. 25-43, 1999.

[Hasselbring, 1999] W. Hasselbring, "Top-down vs. Bottom-up engineering of federated information systems", in *Proceedings of EFIS'99*, pp. 131-138, Infix Verlag, 1999.

[Hick, 1999] J-M. Hick, V. Englebert, J. Henrard, D. Roland, J-L. Hainaut, "The DB-MAIN Database Engineering CASE Tool (version 5) - Functions Overview", *DB-MAIN Technical manual,* Institut d'informatique, University of Namur, November 1999.

[Parent, 1998] C. Parent and S. Spaccapietra, "Issues and Approaches of Database Integration", *Communications of the ACM*, 41(5), pp.166-178, 1998.

[Papakonstantinou, 1995] Y. Papakonstantinou, A. Gupta, H. Garcia-Molina, J. Ullman, "A Query Translation Scheme for Rapid Implementation of Wrappers", *International Conference on Deductive and Object-Oriented Databases*, 1995.

[Ramesh, 1995] V. Ramesh and S. Ram "A methodology for interschema relationship identification in heterogeneous databases, *Proceedings of the Hawaii International Conference on Systems and Sciences,* pp. 263-272, 1995.

[Schwarz, 1999] K. Schwarz, I. Schmitt, C. Türker, M. Höding, E. Hildebrandt, S. Balko, S. Conrad, G. Saake, "Design Support for Database Federations", in *proceedings of ER'99*, Paris, November 1999.

[Sheck, 1991] H-J. Sheck, G. Weikum, W. Schaad, "A Multi-level Transaction Approach to Federated DBMS Transaction Management" in *Proc. 1st Workshop on Interoperability of Multidatabase Systems*, pp. 280-287, IEEE Computer Society Press, 1991.

[Sheth, 1990] A.P. Sheth and J.A. Larson “Federated Database Systems for Managing Distributed, Heterogeneous and Autonomous Databases”, *ACM Computing Surveys*, 22(3):183-236, September 1990.

[Sutherland, 1997] J. Sutherland, D. Patel, C. Casanave, G. Hollowell, J. Miller, *"Business Object Design and Implementation"*, Springer-Verlag, Berlin, 1997.

[Thiran, 1998] Ph. Thiran, J-L. Hainaut, S. Bodart, A. Deflorenne, J-M. Hick, “Interoperation of Independent, Heterogeneous and Distributed Databases. Methodology and CASE Support: the InterDB Approach” in *Proceedings of CoopIS’98*, IEEE, New-York, August 1998.

[Thiran, 1999] Ph. Thiran, J-L. Hainaut, J-M. Hick, A. Chougrani, "Generation of Conceptual Wrappers for Legacy Databases", in *Proceedings of DEXA'99*, LCNS, Springer-Verlag, September 1999.

[van den Heuvel, 2000] W.J. van den Heuvel, W. Hasselbring, M. Papazoglou, "Top-Down Enterprise Application Integration with Reference Models" in *Proceedings of EFIS'00*, pp. 11-22, IOS Press and Infix, 2000.

[Vermeer, 1996] M.W.W. Vermeer and P.M.G Apers, “On the Applicability of Schema Integration Techniques to Database Interoperation”, in *Proc. Of 15th Int. Conf. On Conceptual Modeling*, ER’96, Cottbus, pp. 179-194, Oct. 1996.

[Wiederhold, 1992] G. Wiederhold, “Mediators in the Architecture of Future Information Systems”, *IEEE Computer*, pp. 38-49, March 1992.

[Wiederhold, 1995] G. Wiederhold, “Value-Added Mediation in Large-Scale Information Systems”, *IFIP Data Semantics (DS-6)*, Atlanta, Georgia, 1995.