

## Data Interoperability: Standardization or Mediation

Scott A. Renner  
Arnon S. Rosenthal  
James G. Scarano

The MITRE Corporation

### 1. DATA INTEROPERABILITY: THE PROBLEM

The data heterogeneity problem is well known, and is illustrated in Figure 1. There are five different data schemas, all representing the same information about aircraft maintenance schedules. Systems A and B are different only in the names of the data elements. Systems B, C, and D have a different structure: the type of aircraft is represented as a value in system B, an attribute name in system C, and a table name in system D. Finally, systems A and E are structurally equivalent, but differ in their representation for ACTYPE and the scheduled completion time. In order for any of these systems to exchange information, the data must be converted from the source schema to the receiver schema.

---

System A			System B			System C		
Table: ACMAINT			Table: RDYACFT			Table: MAINTSCHED		
<u>ACTYPE</u>	<u>RDYWHEN</u>	<u>NUM</u>	<u>MODEL</u>	<u>AVAILTIME</u>	<u>QTY</u>	<u>RDYTIME</u>	<u>F15S</u>	<u>F16S</u>
F15	0500	22	F15	0500	22	0500	22	-
F16	1700	16	F16	1700	16	1700	-	16
System D			System E					
Table: RDYF15S		Table: RDYF16S		Table: ACMAINT				
<u>WHEN</u>	<u>QUANTITY</u>	<u>WHEN</u>	<u>QUANTITY</u>	<u>ACTYPE</u>	<u>RDYWHEN</u>	<u>NUM</u>		
0500	22	1700	16	F-15	5:00A	22		
				F-16	5:00P	16		

---

Figure 1: Same information, five separate data schemas

(A related problem, and one which we do not attempt to solve, is that the same *data* may not be the same *information* in the source and receiving system. For example, it does no good to have a complete match in name, structure, and representation of data, if RDYWHEN in one system tells the time repairs will be complete and in the other tells when the aircraft should be positioned for takeoff. All we can do with this kind of semantic mismatch is try to detect and avoid it.)

The current approach to the data interoperability problem is to write *ad-hoc* data interface programs for each pair of communicating systems. Experience shows that development and maintenance of these programs is expensive in terms of both time and money. Worse, the total effort required increases with the *square* of the number of communicating systems. Finally, these hard-coded interfaces support only the information transfer anticipated during development, and not “pull-on-demand” transfers. It is plainly evident that the current approach cannot be made to support the requirements of the infosphere, or even those of the immediate future.

## **2. DATA STANDARDIZATION IS NOT THE (WHOLE) SOLUTION**

If every system always used the same data to represent the same information – identical names, structure, and representations – then the data interoperability problem would go away. The DOD data standardization program will do this to some extent, but there are reasons why standardization will not be a complete solution, which we will consider in this section. Similar arguments would apply to other organizations (e.g., NASA, or the oceanography community) that try to impose a single, monolithic standard.

Constructing and maintaining a single, integrated standard data model is difficult or impossible.

For every system to use the same data to represent the same information, we need a single, integrated data model covering the union of the system domains. This approach can work for small, simple enterprises. However, the DOD is neither small nor simple. Purely from the standpoint of human limits on comprehension, we should not expect success in constructing a single model of appropriate detail for a large enterprise. Instead, we should expect many models, each covering a single functional domain. Systems will adopt data definitions from the appropriate model. This introduces data interoperability problems wherever systems communicate across the boundaries of separate models.

The standard will change, but systems will not all simultaneously change to conform.

Even if we somehow obtained a single data model, change is inevitable: as the world changes, so must our representations of it. In large-scale models, these changes will be frequent. For example, comparatively stable databases might require one schema change every three years. A standard model covering 100 such databases must cope with a change every two weeks. [Goh94]

Simply to plan a transition, one needs the ability to accommodate two conflicting sets of metadata, and to use that metadata to generate data translation routines. Worse, it is often impossible for all systems to coordinate so that changes take effect simultaneously. Instead, systems will come into compliance over time. This introduces data interoperability problems between the systems that have changed and those that have not.

There will always be a requirement to communicate with non-conforming systems.

No data standard is of any use when your communication partners have not adopted it. We believe that systems will always be required to exchange information with systems that do not conform their favorite data standard. We do not expect that allied systems (e.g. the French army) and commercial systems (e.g., Federal Express, United Airlines, Wal-Mart) will be adopting the DOD data standard any time soon. We also believe that some DOD legacy systems will be around much longer than many people expect. Information exchange with these systems introduces all of the data interoperability problems mentioned above.

Similar difficulties arise in scientific fields. . For weather modeling, it may be necessary to cross disciplines (meteorology, oceanography), organizations (NOAA, Navy, NASA), and national lines.

Certain representation choices resist standardization.

A key notion in data element standardization is that one fully defines the semantics *and representation* for each attribute of each entity type. (More precisely, one has a library of *data element* definitions, and each attribute is bound to one of those definitions). That is, all of the properties of the data element are known and standardized [DOD93]. This facilitates information exchange between systems that use a standard data element. No translations are required, because the name and representations are known to be the same.

We believe that there are legitimate reasons why systems might need different representations for the same attribute. Imagine two systems which keep track of the time at which some event is observed. System A makes its observations with a very precise electronic sensor, while system B receives event times from a variety of automated and human observers. Within the context of the data model, the two systems are recording information about the same fact, so we would like them to use the same data element. . However, any attempt to define a standard precision would be destructive.

There are three possible solutions to this problem. First, we can ignore it, by causing both systems to use a data element defined at the higher precision. System B simply ignores the extra precision digits in its database, probably filling them in with meaningless zeroes. This amounts to telling a lie about the data. The trouble is that sooner or later, the lie is going to be believed. Some system will receive system B data and treat it according to the high precision that it claims, not the low precision that it actually has.

A second solution is to introduce new data elements to explicitly represent the different metadata. That is, in addition to the time-of-event data element, the data model could have a precision-of-time-of-event and a confidence-in-time-of-event data element. This has a good chance of avoiding the probable data-accuracy error in the first solution. However, it causes

the two databases to be filled with data that neither system actually needs. (For example, in system A, the precision and confidence data elements *always* have the value “high.”)

The third solution is to create separate, fully-defined data elements for each combination of metadata. Systems A and B may then have distinct, properly-defined data elements for recording their time-of-event data. We avoid the data-accuracy error and avoid introducing unnecessary data elements. However, this approach may greatly increase the number of data elements in the data standard. If there are five possible precisions and five levels of confidence, then we need 25 “standard” data elements just for the time-of-event concept. Also, we introduce data interoperability problems. For information to flow from A to B, we need a human to notice the similarity in data elements and to supply the appropriate translation between them.

None of the above solutions is particularly attractive. We believe that a better approach is to relax the fully-specified-metadata requirement, permit system developers to specify the metadata used in their data schema, and provide tools for automatically resolving metadata differences. We will return to this theme in section 3.

### 3. DATA MEDIATION PROVIDES A PARTIAL SOLUTION

We are developing a *data mediation* approach to solve the data interoperability problem. A data mediator is a computer program which translates data between two systems with different data schemas. In our approach, the mediator handles an information exchange between a source and receiver system in two steps. Beginning with a query from the receiver’s schema, we first translate it into the equivalent query against the source schema. Then, we execute the source query and translate the retrieved source data into the receiver’s format. The result is that the mediator acts as a *semantic gateway* between the systems, permitting the receiver to view the source as an extension of its own database, without concern for the differences in names and representations of data.

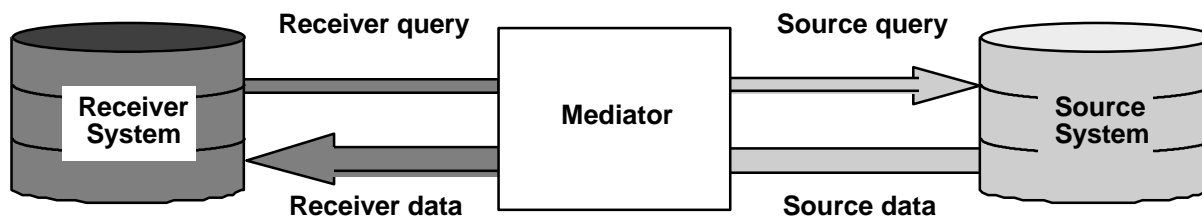


Figure 2: Data mediation

The difference between a mediator and the hard-coded translators in use today is that the mediator automatically generates data translations from descriptions of the data in the source and receiver schemas. These descriptions must have the following properties:

- They must be written using a *formal language*. Natural language descriptions may be sufficient for human developers, but not for the mediator program.
- They must be written using a *common vocabulary*. System developers must agree on the meanings of the terms in their descriptions in order for the mediator to identify and correlate related elements in different schemas.
- They must be *adequate* for the mediation that is required. (It is not necessary that they be *complete*. Those aspects of a data schema which are never exchanged with another system need not be described at all.)

Our approach depends on a shared, conceptual reference schema (or ontology), as in the Carnot project [Collet91] and the SIMS project [Arens92]. The descriptions of the source and receiver schemas (collectively, the *component schemas*) use the terms defined in the reference schema as their common vocabulary. In our implementation, the reference schema is composed of an IDEF1X data model covering the functional domain plus a library of data element conversion functions. The component schema descriptions are expressed as database views which show the correlations between component and reference entities, plus enough semantic information about each component data element to permit the mediator to select the proper conversion functions.

In addition to the schema correlations, each attribute in the component schemas is annotated with the metadata describing its meaning and properties (e.g. precision, units of measure, quality, etc.) These properties are the *meta-attributes* for the data element, and collectively form the definition of its *semantic domain*. The list of meta-attributes is defined in the reference schema; this supplies a common vocabulary for describing the meaning of data elements. When corresponding data elements in the source and receiver schemas have different semantic domains, the mediator searches its library of conversion functions to compose a sequence of calls which eliminate the differences, as in [Sciore94].

#### **4. DATA MEDIATION FITS WELL WITH EMERGING DATABASE ARCHITECTURES**

Many organizations now wish to support applications that access data in multiple DBMSs. A two layer architecture has achieved widespread acceptance:

- A distributed DBMS provides a uniform query language, and a virtual database to represent data in many physical databases. Roughly speaking, users see a single

database containing all tables from all components, accessible to a single “global” DBMS.

- For each component DBMS, one buys a gateway that translates queries and data between the component’s native format and the D-DBMS format. That is, the gateway makes it appear that the component is running a local copy of the global DBMS.

Our mediation approach is quite compatible with this architecture. As shown in the next figure, we add a *semantic gateway* for each component database. Just as the DBMS gateway above made it look like each component natively used the global DBMS, the semantic gateway makes it look as if there is a global schema, and each component is using it.

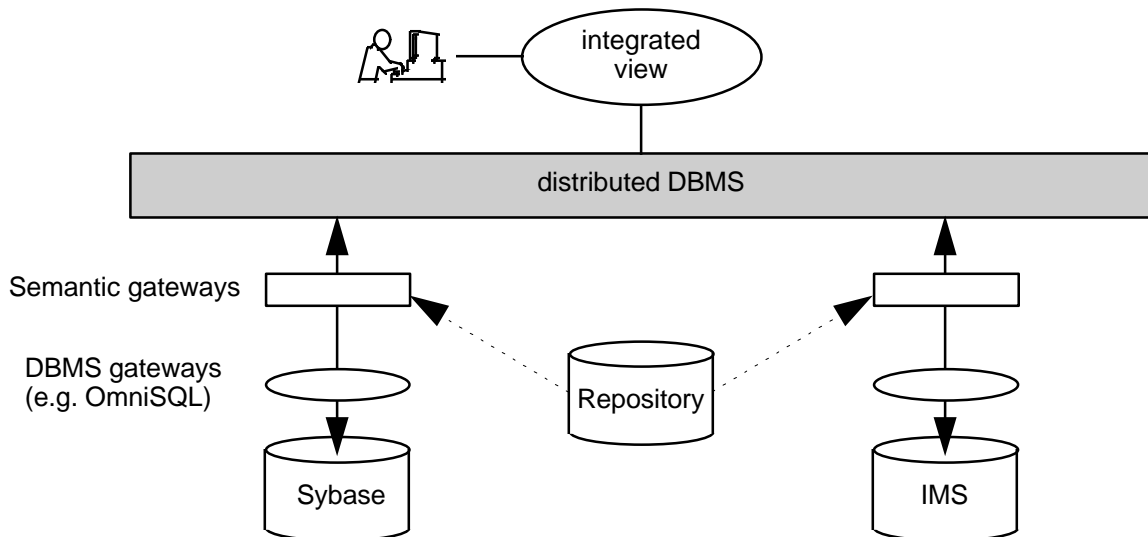


Figure 3: Data and DBMS Heterogeneity ó Two Kinds of Gateways

## 5. ADVANTAGES OF DATA MEDIATION

We believe that our approach is complementary to the DOD data standardization program. Data standardization will reduce the need for mediation, but will not eliminate it. The standard data models developed as part of the 8320.1 process can serve as the backbone of the reference schemas required in our mediation approach. We expect that in time we may identify some additional metadata that should be collected for standard data elements.

The component schema descriptions produced for our approach are *reusable*. Developers describe their data schema once; the description can then be used by the mediator to

communicate with as many other systems as required. This solves the interface explosion problem. It does not matter how many data interface programs are generated by the mediator; the work done by developers — writing descriptions of their data schemas— grows linearly with the number of communicating systems.

The component schema descriptions are easier to write and maintain than the equivalent interface code. It is less work to write  $n$  descriptions than to hand-code  $n^2$  interfaces. We also believe that it is simpler, and more efficient to write and maintain data knowledge in a declarative description, than to maintain the same knowledge embedded in the procedural source code of a data interface.

The component schema descriptions can serve as precise, reliable documentation for system developers and users. The traditional system of free-text comments is notorious for outdated, imprecise documentation. Because our descriptions are used to produce system software, they must be kept current. Because our descriptions are the input to a computer program, their meaning must be precisely defined.

Data mediation allows individual systems to keep their own “view of the world.” Users are typically reluctant to abandon their own data schema in favor of a standard schema supplied by someone else. The infosphere is supposed to supply information to users in the form they require; data mediation allows users to specify “what they need and how they need it.” Our approach allows users to keep their schema so long as they can describe it to the interface generator. Our approach is 100% “carrot” — use the mediator, because it makes your information exchange tasks simpler. Data standardization is 100% “stick” — adopt the data standard, or (eventually) lose your program funding.

Unlike many other proposals for integrating heterogeneous databases, our approach works as a layer on top of the existing DBMS systems. It does not require a new query language, or the installation of new, leading-edge DBMS software. We are concentrating now on relational database systems. However, we believe that our approach can be extended to work for other database models, as well as for generating standard message format interfaces.

## 6. SUMMARY

Data interoperability is a problem for C<sup>3</sup>I systems now, and will remain a problem in the future. Data standardization helps, but is not the whole solution. We are developing a data mediation approach which complements the data standardization process, supplies the missing parts of the solution, and can be integrated with existing C<sup>3</sup>I database systems. We believe that our data mediation approach will provide a flexible, cost-effective mechanism for satisfying the information exchange requirements of all types of systems.

## 7. REFERENCES

[Arens91]

Arens, Y., and Knoblock, C. A. Planning and reformulating queries for semantically-modeled multidatabase systems. In *Proceedings of the 1st International Conference on Information and Knowledge Management* (1992), pp. 92-101.

[Collet91]

Collet, C., Huhns, M. N., and Shen, W. M. Resource integration using a large knowledge base in Carnot. *IEEE Computer*, Vol. 24, No. 12, December 1991.

[DOD93]

Department of Defense, *Data Element Standardization Procedures*, January 1993. DOD 8320.1-M-1.

[Goh94]

Goh, C., Madnick, S., and Siegel, M. Context interchange: overcoming the challenges of large-scale interoperable database systems in a dynamic environment. In *Proceedings of the Third International Conference on Information and Knowledge Management* (1994), pp. 337-346.

[JCS92]

Joint Chiefs of Staff, *C4I for the Warrior Objective Concept*, September 1992. Coordination draft.

[Sciore94]

Sciore, E., Siegel, M., Rosenthal, A. Using semantic values to facility interoperability among heterogeneous information systems. *ACM Transactions on Database Systems*, June 1994.

**Acknowledgement:** Some of this material has been presented at the DOD Database Colloquium '95, San Diego, August 1995.