

## **37. Mediators, Concepts and Practice**

### ***Gio Wiederhold***

Prof. Emeritus, Stanford University

[Gio@cs.stanford.edu](mailto:Gio@cs.stanford.edu)

for Joachim Hammer (editor) Handbook of databases.

1 June 2007

### **37.0 Abstract**

Mediators are modules in large-scale information systems that link multiple sources of information to applications. Mediated systems exploit autonomous services to permit growth and enable them to survive in a semantically diverse and rapidly changing world. Constraints of scope are placed on mediators to assure effective and maintainable systems. Modularity in mediated architectures is not only a goal, but also enables the goal to be reached.

### **37.1 Overview**

We first present the role of mediators and the architecture of mediated systems, as well as some definition for terms used throughout this exposition. Section 3 deal with mediators at a conceptual level. Section 4 presents the basic functionalities, and Section 5 present the primary role of mediators, information integration, including the problems of heterogeneous semantics, and the modeling of knowledge to drive integration. Section 6 points to related topics, not covered as such in this chapter deals. A final summary reviews the sate of the technology in 2006, indicating where research is needed so that the concepts will scalably support information systems.

#### **37.1.1 Architecture**

Mediators interpose services in large-scale information systems to support applications used by decision-makers, where the scale, diversity, and complexity, of relevant data and information resources are such that the applications would be overwhelmed.

Augmenting databases and other base information sources with adequate functionality is equally impractical, particularly because multiple, autonomous resources cannot support all the possible combinations of application requirements. Figure 37.1 sketches the basic layering of mediation. More complex layering can be envisaged, but it is useful to keep this exposition simple.

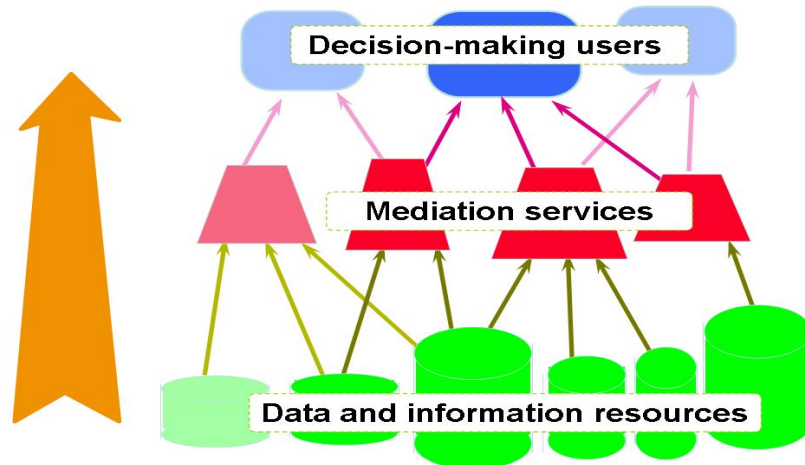


Figure 37.1 Place and role of mediation.

The mediator architecture envisages a partitioning of resources and services in two dimensions [Wiederhold:92]:

- Horizontally into three layers: the client applications, the intermediate service modules, and the base servers.
- Vertically into multiple domains or areas of expertise.

The result is a modularization which enables effective maintenance. For a specific application the number of domains and for each domain the number of supporting servers is best limited to  $7 \pm 2$  [Miller:56], allowing applications to easily obtain access to dozens of sources, while the components can participate in a much larger network.

### 37.1.2 Motivation

Today, few large information systems are built from the ground up. Instead, they are constructed from existing components and systems. Systems integrators supply only functionalities necessary to make the pieces work together. Integration can take place at several levels of granularity. Composition of basic software modules is the approach used in object-oriented (OO) software engineering. Composing webservices is the approach when the modules are remote, and perhaps owned by other parties, but where standards assure compatibility. Middleware resolves format incompatibilities, providing transformation among differing standards, including proprietary conventions. Mediation attacks the next level of compositional scale, where large systems provide services, but these services, being independently developed, will not have been designed as components, and hence cannot be expected to be compatible.

While middleware resolves technical interface issues that arise between services and their users, the intent of mediators is to deal with semantic differences. Semantic differences are hard to resolve. Standards, where they exist, depend again on terminologies that are hard to pin down. Only when the definitions are finally reduced to enumerated real objects can full agreement be assured. But in a world that is ever growing one cannot be limited to existing objects, and one would also lose all the power the use of abstract concepts brings to information systems.

### 37.1.3 Interfaces

For a system composed of modules interfaces are crucial. Mediation requires interfaces at two levels, as sketched on Figure 37.2. Information technology has a surfeit of interface standards, and we cannot cover specifics in this exposition. Today, XML is the prime candidate for delivery of information to an application [Connolly:97]. Its hierarchical structure supports a useful model, as described in Section 35.5.3. Early applications often used CORBA [OMG:91]. It is important to note that there is no need for a user-friendly interface, instead we need at this level a machine- and communication-friendly interface.

For obtaining information from the data resources one has to adapt to what is available: SQL, XML, OQL, search engines, data mining programs, etc. [Kim:95]. Wrappers are needed to achieve commonality.

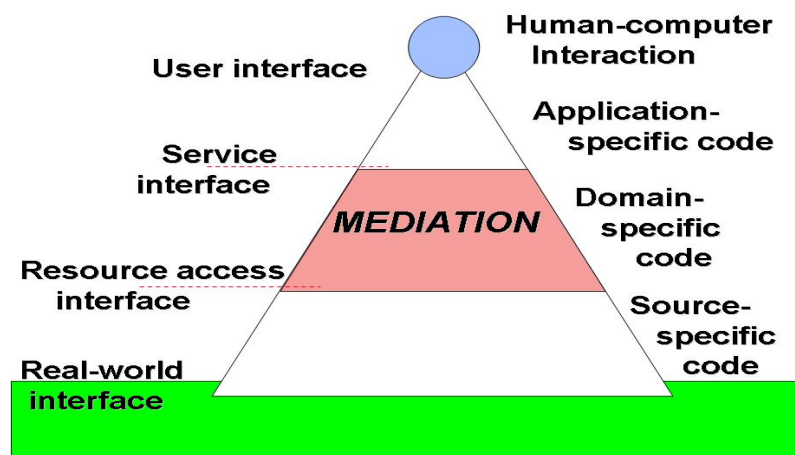


Figure 37.2 Interfaces

By formalizing and implementing mediation a partitioned information systems architecture is created. The modules are of manageable complexity and in combination can deliver the power that technology makes available. The partitions and modules map into the distributed hardware concepts of servers and webservicees.

### 37.1.4 Complex architectures

Structuring mediators into hierarchies should not lead to problems. Low-level mediators may only have database access knowledge, and understand little application domain semantics. High-level mediators may take on well-defined decision-making functions. There is also a role for search mediators that can inspect and propose mediators for application use. Such service discovery is envisaged for web-based service, but even within a domain assuring that the required metadata is available is a daunting problem [CaceresFOV:06].

In general, it is wise to gain experience with simple architectures, before starting to generalize and create layers that make systems hard to understand. Keeping the structure of any application system seems wise. In many application domains experts and human

agents are available that have performed the task manually. Just as human organizations are willing to structure and constrain interactions, even at some lost-opportunity cost, it is wise to impose similar constraints on the broad information systems we wish to deploy.

## **37.2 From Facts to Knowledge**

We must introduce some definitions for this chapter, since the world of information technology (IT) has become so broad that identical terms are used - and misused - in a variety of contexts, often obvious to the reader, but confusing when trying to build semantic bridges.

**Databases** are best viewed as retainers of observed and recorded facts. Ideally each data element should be verifiable by comparing it with a real-world artifact. In practice observations and recording is imprecise, often occurring in the past, so that validation becomes impossible. Having stored data is a prerequisite for information processing, but is not sufficient.

Data should be verifiable by observing the real world. But the real world may well have changed since the observation was made and recorded. Ideally any data element should have a timestamp associated with it [BöhlenJ:07]. But that ideal is rarely reached.

Most databases also contain summary data, perhaps because the volume of all the detail was overwhelming, or computed from detail, to aid the prime user of the database. Since a single source database would live within a single context, there should be few problems due to semantics, and we can accept all if its contents as facts.

There are databases that have been built by combining data from disparate databases. We regard them as less trustworthy, as was shown using some analyses we performed on the CIA factbook. The convenience of using such aggregated databases - everything you want to know is in one place, is offset by errors due to temporal inconsistencies, changes in classification, and simplistic semantics [WJ:98]. We will cite some problems encountered later as examples of mediation tasks.

**Information** is created when data are being processed, such processing requires knowledge. Having knowledge is primarily a human capability, but can be encoded in programs. Common operations are summarizations, here knowledge is needed about the structure of that data. Should facts about sales be aggregated by location, by type, by producer, or by customer? What is information is hence determined by the receiver. The technical definition of information [ShannonW:48] is yet more narrow: information should be novel to the reader, otherwise it is of no import.

Indeed, stored information does little good until it is transmitted to a reader. The reader can absorb that information, it should add to the users' knowledge, and perhaps cause the reader, now in the role of a decision maker, to initiate some action.

Information can also be further processed, creating information further removed from the source of the data and subjected to more assumptions, based on the knowledge of any

intermediaries. The representation of information will change once it is processed, typically becoming more complex and less amenable to simple processing algorithms. Information in the form of written text, such as this article, is very far removed from the many sources, and even though we try to validate it by giving references, tracking all the real world facts that contributed to it is well-nigh impossible, although a topic of current research [MutsuzakiEa:07]. The intent of this article is to contribute to your knowledge, and it depends on your prior knowledge how much of the information will increase your knowledge.

**Knowledge** is required to process data or information, be it extracted from a database or transformed into text. For human processing the limits are education, and to some extent the ability to profit from education, those qualities that makes people knowledgeable. Programs are the prime means for representing knowledge in computing, but maintaining knowledge in that form is painful. The choice of mechanical representation for knowledge in computing form remains a prime issue of artificial-intelligence research, it is inextricably bound to the processing algorithms that can operate on them. Mediators embody knowledge to carry out their functions. It appears that in many cases much of the knowledge would be easy to represent.

**Actionable information** is information that actually causes a decision-maker to carry out an action, as purchasing something, deciding which alternative to chose in an operation, investing in a new venture, or even abandoning an ongoing project. It is only when information becomes actionable, i.e., does more than increase one's knowledge that economic value is generated. Of course, without having previously received information that increased the decision-maker's knowledge the capability to understand the crucial increment of information may not exist, and actionable information will be lost.

We summarize these definitions below:

<b>Concept</b>	<b>Definition</b>
Data	Recorded Facts about the state of the work
Information	Data or processed data not currently known to the recipient
Knowledge	Personal or encoded information that can drive processing of data or information
Actionable information	Information, that when combined with knowledge can be used to change the state of the world

These definitions cannot cover all facets of information in all contexts. In some settings direct access to fact-based data does not exist. In such a setting low-level information is treated as data.

Since people have a limited capacity to retain knowledge, some information will just help them to recall what they should have known. In general it is easier for us to record

knowledge about facts we don't want to lose as information, and then trust that our processing capabilities will reconstruct the knowledge.

There is also information that amuses, and as such is worthwhile. Being able to recall useless knowledge can be source of pride. But, for a business information should be potentially actionable, even while it increases one's general knowledge.

There are also loops to be closed. Actionable information leads to actions, say a purchase of a piece of hardware. That action changes the facts in the world: after some time there is one fewer pieces of hardware in the store, and one more in your hands. That fact can be recorded, and becomes part of the data cycle. The action will also increase your knowledge, immediately because it validates the purchase process you constructed, and over the longer term, as the new piece of hardware satisfies your expectations and understanding of its catalog description.

In a simple business environment, actionable information is easy to recognize.

- A factory manager needs sales data to set production levels.
- A sales manager needs demographic information to project future sales.
- A customer wants price and quality information to make purchase choices.
- A manager of a healthcare plan has to balance investments in preventive, urgent, episodic, and palliative care.

Most of the information needed by the people in these examples can be represented by factual data and should be available on some computer somewhere. Communication networks can make data available wherever it is needed. However, to make the decisions, it must be transformed to manageable volumes of actionable information, a considerable amount of knowledge has to be applied as well. Today, most knowledge resides within the administrative and technical staff in an institution, and human-mediated intermediate steps are interposed between the databases and the decision makers [Waldrop:84].

### ***37.3 Conceptual principles***

Knowing that information exists in the myriad of resources available on the Internet creates high expectations by end-users. Finding that it is not available in a useful form or that it cannot be combined with other data creates confusion and frustration. The task of mediators is to provide functionalities that extract actionable information for those resources.

Mediators embody in software functionality that selects, transforms, aggregates, integrates, and summarizes data from multiple sources, in order to support information needs of decision makers.

While this is an ambitious statement there are simplifying conditions that make the creation and maintenance of mediators feasible.

### **37.3.1. One-directional flow**

We expect mediators only to process data in one direction, towards the end users. Since we assume that the sources are truly independent, there is no authority here to alter them. If inconsistencies are found they will be reported to the decision-making programs. Mediators may include knowledge about the credibility of the sources, and uses such information to select or weigh the information for aggregation and summarization.

It is up to the receivers to provide feedback to inconsistent sources, if that is desired. Often inconsistencies are natural, and no feedback is needed. For instance, if one source obtains more recent data, it will be preferred, and only major inconsistencies will be of concern.

### **37.3.2. Delegation of technical incompatibilities.**

So that mediators, and their creators and maintainers, can concentrate on issues at the semantic level matching to transmission and representation standards is carried out by lower-level modules, i.e., middleware [Bernstein:96] or specialized wrappers [HammerGNYBV:97]. Textual data especially requires processing that is best delegated [IpeirotisAJG:06].

Note that these low-level modules need only be concerned with a one-way match of two interfaces, that of a source and that of the mediator. Where multiple sources have identical interfaces those modules can be reused or replicated. There is no need to consider n-way interaction at this lower level.

### **37.3.3. Constraint to commensurate semantics.**

Mediators must carry out reliable computation, so that they can be trusted by the decision maker. That means that data that are inherently not comparable cannot be included in one mediator. An example in healthcare would be cost of patient services and quality of patient services. We expect that in this case two mediators would be required, one managed by the CFO of the institution, that aggregates detail and presents them as cost per type of patient, and one by the chief medical officer, who aggregates the variables that are indicators of quality of care. It is the decision-maker who receives both types of data and must make the difficult choices in balancing the two. If that task could be automated a third mediator would need to model the decision-maker's reasoning.

The number of mediators required to serve a decision-making application can hence vary from one to many. One suffices if all the information being obtained from the sources can be reduced to common units. The maximum number would be small in practice, since few decision-makers will be able to balance more than several incommensurate inputs. In complex situations there might be hierarchies of decision makers, each receiving input from multiple mediators, perhaps sharing the use of some of them. We have not seen practical examples of such hierarchies.

### **37.3.4. No requirement for automation.**

No requirements for automatic generation or maintenance are imposed on mediator technology. Having automatic discovery, linking to sources, and adaptation when sources change to facilitate mediation is exciting, but hard. Such automation is the goal of artificial intelligence research, and as progress is made in that arena, that technology should transition [WiederholdG:97]. But gaining experience with a new technology is essential for learning how to automate it, and such experience is gained by actual building and maintaining mediators.

Much of the logic in mediators can be placed into tables, simplifying ongoing maintenance. In Section 5.5 of this chapter we discuss what type of maintenance will be required.

### **37.3.5 What is left?**

These four conditions limit the issues that mediators still have to deal with, but a central, very hard issue remains: the conversion of voluminous data from many disparate and autonomous sources into integrated actionable information. Autonomy implies that no standards enforcing consistency of representation, nor content, nor abstraction level, nor vocabulary can be imposed. But having systems in place that will benefit from consistent semantics will encourage participants, over time, to either adopt standards or document the formats, intent, and scope of their data.

Even those four conditions listed are not cast in silicon. Many researchers and implementers have gone beyond the conditions. But, in the process of covering more, many projects have not covered the basic functionalities that are needed to make mediation effective.

### **Mediators versus data warehousing.**

An alternative data integration technology is data warehousing [Jarke:07]. Warehouses collect data from multiple sources, and will store large volumes. The warehouses maintainers cannot worry greatly about semantic consistency. Submitting queries to a warehouse does not require the intermediate processing expected to occur in mediators. However, keeping a warehouse complete and up-to-date requires costly ongoing maintenance, since every change in a contributing database should be reflected in the warehouse. This cost limits the scope of warehouses in practice. Mediators can hence cover a wider range of sources than warehouses, but pay for that by slower execution times.

Conceptually, mediators select the required intersection of the source information, while warehouses provide the union. While mediators are hence quite distinct from data warehousing, it seems feasible to design mixed systems, using warehousing technology for relatively static data and mediators for dynamic information. Mediator queries can be treated similar to warehouse view queries, if the warehouse is consistent [Ullman:00].

## 37.4 Operations and Managing Volume

A decision-maker needs a small amount of high-quality actionable information. Just making the underlying sources available in an attempt to produce 'all the information' cause *information overload*. The role of the mediators is to greatly reduce the volume that arrives at the decision maker, while losing little of the information the decision-maker needs.

Reductions in volume are made by

1. Selection of relevant data records to be obtained from the sources.
2. Projection, i.e., selection of relevant columns of the data.
3. Aggregating the selected data to the level of abstraction needed for integration.
4. Reducing the integrated and aggregated data further.
5. Pruning the results to reduce application overload.

If the sources can handle queries at the complexity of SQL, then reduction tasks 1, 2, and 3, can be combined, and the mediator itself will have less data to cope with. For less competent sources reduction is limited to selection and retrieval of relevant records. It is the task of the interface to the sources to inform the creator of the mediator what functionalities are available. For automation, that information must be in machine-processable form, as discussed in Section 5.3 of this chapter.

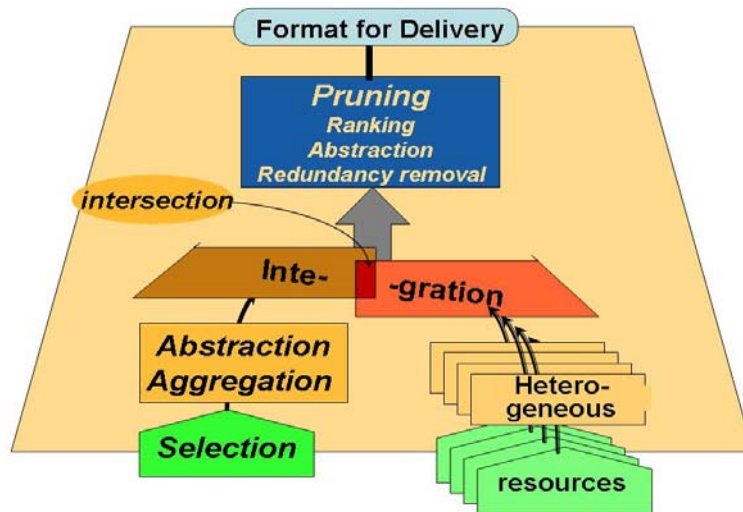


Figure 37.3 Functionalities within a mediator

### 37.4.1 Selection and Projection

It is not coincidental that SELECT is the principal operation of relational database management systems, since this the most important functionality. Getting only the relevant rows, or in OO terms, the relevant objects, typically reduces volume by several orders of magnitude, especially for voluminous sources. Sometimes an adequate tactic for providing the required information could be by sampling [Olken:86]. To select from textual sources, selection requires exploiting indexes, limiting retrieval to records where terms intersect, and often relevancy metrics to further reduce the volume [FriederG:07].

Projection reduces the number of columns or the number of attributes. In a relational setting projection can also reduce the number of rows if key attributes are projected out. Getting rid of key attributes is rare though for early stages of mediating processing, since they are often needed for integration. Often data not needed for the decision maker must be retained to enable integration. In the end projection may reduce volume by an order of magnitude.

### **37.4.3 Aggregation.**

Selected data is typically at too fine a level of detail to be useful for decision making. Especially when the sources are operational databases, where every transaction is logged to assure completeness, the amount of detail is overwhelming. Distinct sources may also provide data at different abstraction levels, say sales data by store from one source and income level by municipality from another source. Aggregation from store to municipality of the data from the first source will be needed prior to combining the information. The central issue of integration will be described in its own Section 37.5 below.

Aggregation operations as {COUNT, AVERAGE, SD, MAX, MIN}, etc. provide within SQL computational facilities for abstraction, and can reduce the volume of input to the mediator. Often adequate query capabilities do not exist and the abstraction must be computed within the mediator. For instance, abstractions that require recursive closures cannot be specified with current database query languages. A classic, although trivial, example is finding one's grandfather via transitivity of parents.

Temporal aggregations, say combining weekly data into monthly summaries are also beyond current database query capabilities [BöhlenJ:07]. If sales data is to be aggregated, corrections may be needed to adjust for the number of weekends in a specific month. Geographic aggregations present similar problems, say information coded by postal-code versus town-limits [Wolfson:07].

Common computations which convert data to a more useful abstraction involve summarization over hierarchies, changing temporal granularity, performing seasonal adjustments, enumerating exceptions, and recursive path computations. Diverse mediator modules will use these functions in various combinations to provide the support for user applications at the decision-making layer.

### **37.4.4 Abstraction**

Abstraction is the process of reducing data to more meaningful units. Abstraction is often required prior to integration, since source data, collected in distinct base systems, will likely differ in representation. After integration, data should be aggregated and abstracted according to the user's information needs

Numeric measures are often too specific to serve decision-making and also are hard to match during integration. It is unlikely that even very similar data will match. Non-

essential differences may even due to the precision used or the choice of metrics, as 3.14 versus 3.1415926535897932384626433832795, or 1 mile versus 1.609 km. In general, numeric data is converted to categories, as tiny, small, medium, large, huge. In general, 7 categories provide adequate distinctions [Miller:56]. Categorization reduces volume greatly, but can rarely be requested to be computed within the sources, so that input volume will remain large, and be a major task in mediation.

Category breaks may be based on multiples, as 0, 10, 100, 1000, etc., or based on having categories of roughly equal amounts of contents, as voting districts, or on application significance, as newborns, babies, children, teens, young, middle-aged, senior, old, bedridden, for healthcare. Middle-age may be defined as age from 30 to 60, and comprise most of the population, but the category will be treated similarly. The selection of categories could be an option provided by the application, although in many application domains there are accepted categorizations. In medicine the categorization of subjective observations ranges from 0 to 10, for absent to expected death.

When results are based on historical records, the detail is likely to be voluminous. Detailed historical data often have to be summarized to permit integration, since time intervals have to be mapped. An initial abstraction is to derive temporal interval representations from time-stamped event data. There are two major styles for representing intervals, open and closed, which have to be mapped to match; open intervals support further algebraic operations best [Jajodia:90].

Prior to integration, the largest common interval is the unit to be chosen. After integration further abstractions are useful, as determining and combining intervals over periods of growth or improvement versus periods of loss or deterioration. Those intervals can then be parameterized by length, delta, and variance. The applications can then present the reduced results in graphical form [deZegher:88]. Mediators should not perform the actual conversion to images, to allow applications the choice of presentation.

### **37.4.5 Removing redundancy after integration.**

Integration, described in Section 37.5, brings together information from multiple sources. The result is likely to contain redundant information. There will be information just used for matching data elements. For instance, if integration involved matching on price, only the categories need to be retained and monetary amounts, currency designations, and inflation correction factors can be removed. Identical data can obviously be omitted. Some matching may have been performed that required matching fields based on object identity, say the town names of `Bangalore', now `Bengaluru'. Again only one entry should be reported, the choice would be based on relevance to the decision maker, typically the most recent one should be provided.

The integrated information may be further aggregated. Then many more columns can be omitted, for instance, all town names if the result is a summary of the Indian software industry.

### **37.4.6 Ranking.**

Information for a decision maker can often be ranked, and then results that are ranked low can be omitted in the result, or presented only by specific request. Ranking can reduce volume greatly; a rule of thumb is that a decision-maker should not be presented with more than seven choices [Miller:56]. But such choices should be different in a meaningful way.

For instance, when scheduling travel from Washington, DC to Los Angeles, significantly different alternatives to be presented to a travel application for a given day are:

Alternative S1: UA59: depart IAD 17:10, arrive LAX 19:49.

Alternative S2: UA199: depart IAD 9:25, arrive LAX 11:52.

giving the traveler a choice to get some work done at home or having time to get settled in Los Angeles and avoiding airline food. A poor qualitative difference in travel scheduling is shown by:

Alternative P1: UA59: depart IAD 17:10pm, arrive LAX 19:49.

Alternative P1: AA75: depart IAD 18:00pm, arrive LAX 20:24.

But some travelers may wish alternative rankings, by price, by frequent-flier bonuses, by minimal time in the air, etc. Neither list now includes flights with stop-overs. A ranking by price would list them, and perhaps not show the non-stop flights at all, and also include other airports in the Washington and Los Angeles area. A listing of all possible ways to get from Washington to Los Angeles, via any U.S. cities within a day would be very long

When search engines rank results, they have had no input from the users or their applications. A mediator can receive such directions, and since the volume to be ranked is much less, can compute the ranking to order, and comply with the preferences of the application.

## **37.5 Integration**

Once source data is at a common level it can be integrated. If there are no semantic mismatches then the data can be combined, typically creating longer records and bigger objects.

At the integration step relational data are often transformed into object or XML formats [Keller:07], [Grust:07]. For complex information the redundancy created by relational join operations or their programmed equivalents can be confusing. If the requesting applications can manage information in object format, such a presentation is a better choice.

### **37.5.1 Heterogeneous Sources**

Much of the benefit from combining distinct sources is that in that process valuable information can be generated, information not actionable from the distinct sources by themselves. However, there is no reason that terms from such distinct sources should match. The terms we must be concerned with are [HalevyIST:03]

1. Terms used in schemas: SQL column names and XML category names

2. Terms used to identify objects, as database keys and XML entry headers, as names, product identifiers, service types
3. Terms used to match data on criteria other than keys, as location, price, quality, etc.

Terms will likely match when the experts that defined the sources have been educated together or have been communicating over a long time. For instance, in medicine, the use of shared textbooks has created a consistent ontology at common levels. But within specialties and recent topics terminology diverge, as in pathology and genetics.

### **Import of heterogeneous semantics.**

In simple retrieval semantic mismatches are often ignored. Synonyms may be employed to assure broad coverage. Search engines leave the resolution of inconsistencies to the reader.

For the business applications that motivate the building of mediators (and warehouses) the occurrence of mismatches creates problems, especially if the results must be delivered to applications that do not have the insight of human readers. A central issue for mediators used in business is then the resolution of heterogeneous semantics.

### **Four Common types of mismatches**

**1. Synonyms** present the simplest problem. The country which is formally the Netherlands is also referred to as Holland. Gambia is listed formally as The Gambia. Name changes can also be seen as synonyms, as Mumbai and Bombay. Simple tables can match these entries. Some matches are harder. The airport for Basel, a Swiss city, is located in France, at Mulhouse. Old documents refer to that town as Mullhausen.

**2. Homonyms**, the use of the same letter sequence for different objects, are the bane of search engines. China is both a country and dinnerware. But the domain constraints in mediation resolve those problems, since relevant sources will be distinct. Attaching the domain or column name to a term can assure keeping them distinct if there is a chance of false matches.

**3. Differences in scope** are best resolved by prior aggregation. For instance information on Czechoslovakia now requires aggregation of the Czech Republic and Slovakia. To avoid losing information an object with subsets for the two current parts may be created. If historical data for Slovakia only is needed, then its 8 lower level regions have to be aggregated when those regions were reported with Czechoslovakia.

**4. Inconsistent overlaps** are the hardest to deal with. A personnel file of a company may list contract consultants that do not appear on the payroll, since they are reimbursed by contract charges. The payroll file may list retired employees, who are paid pensions, but those are no longer part of personnel. Rules to resolve such mismatches depend on the objective of the mediator results. If the average pay of active workers is needed, retired employees should be omitted, but consultants should be included with an equivalent pay rate.

The resolution of such mismatches often requires obtaining data from the sources that will not be needed as part of the delivered results. After integration that data can be omitted, reducing the volume of information delivered to the decision maker. Some sources may only provide data to enable a match, say a table that links salary levels to employee ranking.

### **Futility of fixing the sources**

Integrators often blame the sources for being inconsistent. But those sources have to satisfy their primary objectives, as getting checks into the mail for the payroll, and locating employees that can perform certain tasks, for the personnel file. More insidious are differences in data quality in distinct sources. The payroll file may not keep the employees work location with care, and the personnel file may ignore errors in the social security number. The number of children should be the actual number in the personnel file, but the payroll just keeps a number for tax deductions.

Researchers on webservices have implied that they will be truly successful when all sources become consistent [BernersLeeHL:01]. But such a state is actually not globally desirable. The quality of a source depends on the competency and interest of its maintainers. A project oriented towards billing for healthcare services cannot be relied on to give a full account of diagnoses and clinical problems encountered in a case. The billing personnel cannot be forced to improve their data, and getting clinical personnel involved in improving billing data will be very costly and not help the institution.

In general, it is not feasible to impose on autonomous source reporting of information to a depth that is not within their scope. A blog by a homeowner can talk about what nails were used in a project, perhaps stating their size and shape. A carpenter uses many specific words: sinker, boxnail, brad, etc. Enforcing a common vocabulary is futile and will, in the end, lead to loss of information.

Public data sources often restrict themselves to aggregated data in order to protect privacy of respondents. No recourse exists to fix such databases, although often the data maybe biased or incomplete. When the objective is understood, say to convince people to support some political initiatives, the likely bias has to be taken into account.

### **37.5.2 Mediating Knowledge**

Mediators, because of their focus on commensurate domains and the intersection of the source data, provide an opportunity to deal effectively with heterogeneous semantics. If the application area to be supported already used information from both sources, then there was typically an expert who understood the intersection.

The knowledge of the expert can then be used to devise rules that handle problems due to having synonyms and scope overlap. If data from two sources appear to be redundant an expert will know which source is more trustworthy.

Such rules are best incorporated in rules that can be inspected by all the participants. The maintenance of such knowledge should be assigned as well, best to the specific experts

on the intersecting domains. At times a committee may be required, but in general having a committee slows the process of rule maintenance. If the assignment to a committee is too broad then it is likely that compromises will be made and that precision will be lost. Such loss of precision has been seen in warehouse maintenance, where source heterogeneity is not constrained by domain limits.

Rules devised for mediation can often be validated by going back to the source databases. Applying rules devised to obtain a consistent scope to each of the sources should create a perfect match. If the databases are large, just obtaining counts of the matches provides a validation. Since databases always have errors, having differences on the order of a few percent may not invalidate the rule, but it will be useful to check those exceptions. Sometimes one will find surprisingly large differences. The reason should be tracked down and an additional rule devised. In one case where highway patrol records and vehicle registration records did not match, the cause was that the vehicle registrations included boats. Obtaining another data element and adding a rule to the mediator restricted the match to roadworthy vehicles.

### **Keeping incommensurate information distinct**

For a mediator to be trustworthy it should not try to integrate data that are intrinsically incomparable. In general, balancing cost factors, expressed in monetary units, and quality, expressed in terms of customer satisfaction, should be handled by two distinct mediators. Both can integrate data, sharing some sources, using their own metrics. The decision maker will receive both types of information. Understanding how costs, incurred now, will affect customers' perception of product quality over time, is a task best not automated.

### **Exploiting human capabilities**

To build a mediator, knowledge is needed from diverse sources. A top-level expert may be used to working with abstractions that are not sufficiently precise to allow constructing effective mediators. The required knowledge is strongly related to roles that humans perform in the processing of information:

- a. A technician will know how to select and transfer data from a remote computer to one used for analysis, such information is essential for constructing wrappers.
- b. A data analyst will understand the attributes of the data and define the functions to combine and integrate the data [deMichiel:89].
- c. Agents often deal with integration: A travel agent will be able to match airports and cities, a hardware store customer requests and stock-on-hand, a broker with monetary units and values.
- d. A statistician can provide trustworthy procedures to aggregate data on customers into groups that present distinctive behavior patterns.
- e. A psychologist may provide classification parameters that characterize such groups.
- f. An experienced manager has to assess the validity of the classifications that have been made, forward the information to allow the making of a decision, and assume the risk of that information is adequate to the task.

g. A public relations person may take the information and present it in a manner that can be explained to the stockholders, to whom the risk is eventually distributed. In the process of constructing the mediator, much knowledge is obtained and recorded. It is now available for formal processing and maintenance. The mediator then also provides a corporate memory, one of the goals of corporate knowledge management, but in a focused and more thorough fashion [Holsapple:04].

## **Uncertainty**

Abstraction and integration introduce uncertainty. Some source data, especially if they include projections about the future, are inherently uncertain. Observations and their representations also induce uncertainties. Researchers in artificial intelligence have dealt with many aspects of these issues [HalpernK:04]. A variety of methods to represent uncertainty are available, based on differences in domain semantics. Perhaps all uncertainty computation can be subsumed by probabilistic reasoning [Cheeseman:85] [Horvitz:86]. Uncertainty increases during integration of information [Fagin:99]. Only recently has traditional database research attempted to integrate uncertainty into its processing model [MutsuzakiEa:07].

Mediators should be able to obtain data from sources, use provided or external ancillary data to establish confidence ranges, integrate the results, and provide those results to the decision-makers. Decision-makers are used to operating with uncertainty, since they typically must project the findings into the future, using their knowledge, information from databases, spreadsheets, planning tools, etc. These tools still await effective integration [Wiederhold:02D]. Decision-makers also obtain advice from colleagues, and likely employ some intuition, a reason why the final phase can typically not be automated.

### **37.5.3 Modeling the knowledge in a mediator.**

The knowledge required for mediation can either be explicitly programmed, formulated as rules, or encoded in decision table. The conditions placed on a mediator in Section 37.3 also simplify a formal knowledge representation. Mediators have a specific, application-oriented objective. While unconstrained knowledge requires a complex network representation, the knowledge needed to support a specific task domain can typically be represented in a hierarchical manner, as sketched in Figure 37.4. The application task provides the anchor which becomes the root for the hierarchy and the criteria for subsequent matches [MitraWK:00]. Object oriented technology and XML schemas are adequate to structure the required knowledge.

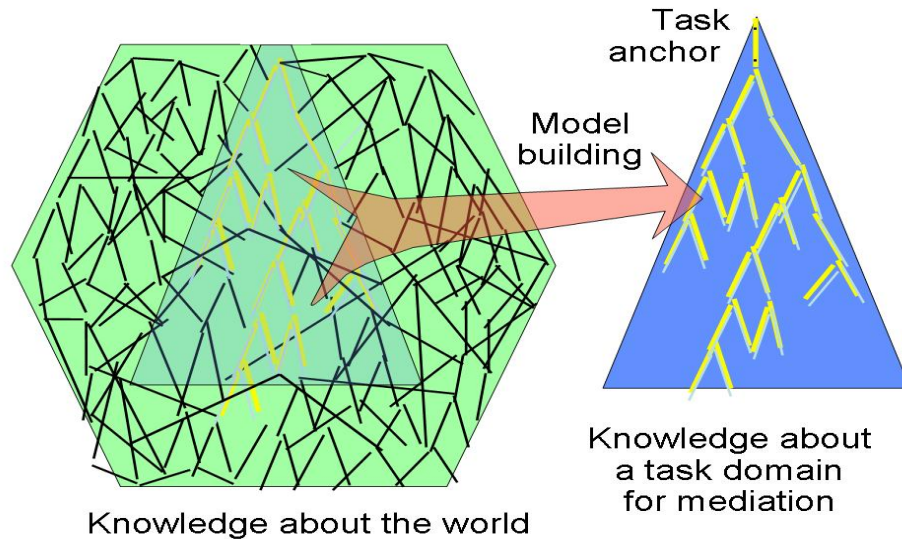


Figure 37.4 Knowledge for task modeling

Much of the knowledge collected when designing a mediator is not formally captured when mediators are implemented by code. Entity-relationship models can provide some formal documentation, but ignore semantic differences [ChenM:89]. The use of rule engines allows representation of rules in a consistent formalism [MalufW:97]. These can then be interpreted by engines as CLIPS [Jackson:99].

Ontologies, if available, provide the meta-data for the description of data and information resources [Guarino:98]. If distinct resources have associated ontologies, the discovery of semantic matches and mismatches among distinct resources can be facilitated and partially automated [StaabSSS:01]. The technologies match those that are used for ontology-based integration [NoyM:00]. For mediation one needs only to obtain the candidate intersections, and human pruning is desirable to limit the number of articulations and avoid unnecessary complexity. As the semantic web develops, more resources will have adequate ontologies, and automation of model-based mediation can make progress [DaviesEa:03].

### 37.5.3 Sharability of mediator knowledge.

The mediator modules will be most effective if they can serve a variety of applications [Hayes-Roth:84]. The applications will compose their tasks as much as possible by acquiring information from the set of available mediators. Unavailable information may motivate the creation of new mediators.

The mediation module which can deal with inflation adjustment can be used by many applications. The mediation which understands postal codes and town names can be used by the post office, delivery services, and corporate mail rooms.

Sharing reinforces the benefits of vertical partitioning into domains. A mediator which only deals with commensurate data can be maintained by an appropriate and trusted expert. The knowledge inserted and maintained in such a mediator will be reused by

many applications. Just as databases are justified by the shared usage they receive, sharable mediators will justify an investment in formally capturing knowledge.

### **37.5.4 Trusting the mediator**

An important, although not essential, requirement on mediators is that they can be inspected by the potential users. Much depends here on the knowledge representation. A coded mediator can have an associated description, but code documentation is notorious for being deficient and poorly maintained. When software is being reused, uncertainties arise, if the original designers made assumptions that will not match the new context.

Having formal models, inherent when rule-systems are in use, will enhance the trustworthiness of a mediator. For instance, the rules used by a mediator using expert system technology can be inspected by a potential [Wick:89]. Still, having access to the human maintainer of the mediator seems to be essential. Providing knowledge for information systems by maintaining may be a viable business, but that has not been proven. The expectation that Internet services should be free hinders the development of quality services that require ongoing expenses.

### **37.5.5 Maintenance**

Our systems must also be able to deal with continuing change. Both data and knowledge change over time because the world changes and because we learn things about our world. Rules that were valid once eventually become riddled with exceptions, and a specialist who does not adapt will find his work to become without value. Any information system must deal explicitly with data and knowledge maintenance.

In mediation the data remains in the resources. But knowledge is required to access, process, and assess those data sources. We know that software maintenance has annual costs of about 15% of the initial investment. While distinct mediators may share software tools, their uniqueness is in the knowledge about the resources and the domain they process. It is likely that mediator maintenance, even if software is shared, will require similar expenditures to maintain that knowledge. Keeping mediating modules small and simple will allow their maintenance to be performed by one expert or at most by a coherent group of experts. In that manner the problems now encountered within large integrated information systems can be ameliorated.

### **Triggers for knowledge maintenance**

Since the knowledge in the mediator must be kept up-to-date, it will be wise for mediators to place triggers or active demons into the databases or their wrappers [Stonebraker:86]. Now the mediators can be informed when the database, and, by extension, the real-world changes. Induction from triggers carries an excessive cost when any state-change must be forwarded through all possible forward chains. For most changes immediate relevance to a user is unlikely. By not carrying induction through to the decision-making layer, but terminating forward chaining in the mediator, that cost can be reduced [Orman:88]. Having intermediate results available in the mediator avoids excessive latency during inquiry. The owner of the mediator should ensure that structural and semantic changes are in time reflected in the mediator's knowledge base.

In a rule-based mediator the certainty factor of some rule can be adjusted. If the uncertainty exceeds a threshold, the mediator can advise its creator, the domain expert, to abandon this rule. The end-user need not get involved [Risch:89].

Eventually mediators may be endowed with learning mechanisms. Feedback for learning may either come from performance measures [Jain:91] or from explicit induction over the databases they manage [Wilkins:87].

## **37.6 Related Topics**

Mediation, just as information technology in general, impinges on many topics of system sciences. We will touch on them briefly in this section, but for discussion in depth other sources must be studied.

### **37.6.1 Private versus public mediation**

Mediation provides a means of portioning information systems by level and by domain. Effective maintenance should be a major benefit, but required maintenance efforts must be assigned and supported. When mediators are constructed, specialists contribute their knowledge about data resources in a manner that applications can effectively share the information. That knowledge may pertain to public or private resources. If the capabilities of the mediators provide a competitive advantage they may well be kept private, even if the mediators access public resources. Such knowledge formalizes the corporate memory, and some mediation technology has in fact been used to capture knowledge of experts that were about to retire.

There may be an incentive for independent specialists to develop powerful, but broadly useful mediators, which can be used by multiple customers. Placing one's knowledge into a mediator will allow rapid exploitation of one's knowledge, and perhaps more rewarding, the writing of a book on the topic.

### **37.6.2 Partitioning versus centralization**

Mediators are oriented towards partitioning of function and knowledge. In that sense they do not follow the database paradigm, where larger often implies better. While warehouses can provide rapid access to massive data collections, they do not deal well with dynamic data and information. And, the knowledge required to understand the data remains outside of the warehouse managers.

Partitioning that knowledge avoids the knowledge centralization, and the associated bureaucracy that ensues when a notion of having corporate information centers is promoted [Atré:86]. It will be impossible to staff a single center, or a single mediator for that matter, with experts that can deal with all the varieties of information that is useful for corporate decision-making.

### **37.6.3 Security and Privacy.**

Mediators gain access to much information, some of which should be protected. The summarization process greatly reduces linkages to individual source data, and can protect

privacy. But assuring that such privacy is protected requires first of all that the mediator, which accesses such data, is kept secure, and that the aggregation is sufficient so that no incriminating identifications escape. Assuring a priori that no results can be used to infer individual source data fatally weakens information processing over irregular data [LinOA:04]. In a mediator dynamic functions can be implemented that analyze results and adapt aggregations to assure privacy protection [Wiederhold:01S] For this function, since a high level of security is required, a distinct mediator should be employed, adding a layer to the architecture.

#### **37.6.4 Efficiency and reliability.**

In actual systems efficiency is always a concern. Each layer in a mediated system should add enough value to overcome the cost of an interface. Standard techniques as caching will be effective where data change less rapidly than application requests. Use of a local warehouse for static data is the ultimate cache. Since information emanating from a mediator has much less volume, the caches can be much smaller than the source information.

In pure mediation every component has to work for the system to work. Mediators can easily be copied and executed at alternate sites. Caches, warehouses, and redundant sources can provide backup when source resources are not available. If the sources cannot be accessed, the delivered data should be identified as being out-of-date. Requirements of data security may impose further constraints. Dealing with trusted mediators, however, may encourage database owners to participate in information sharing to a greater extent than they would if all participants would need to be granted file-level access privileges.

#### **37.7 Summary**

Information systems are becoming available now with the capabilities envisaged by Vannevar Bush for his MEMEX [Bush:45]. We can discover and retrieve documents kept in remote repositories. We can present the values on one of multiple windows. We can select and scroll information on our workstations, we can copy documents into our files, and we can annotate text and graphics. We can reach conclusions based on this evidence and advise others of decisions made.

But actual integration of information, beyond simple data aggregation, as needed for decision-making support, is still uncommon. Most actual decision makers depend on human analysts to provide summaries, aggregate information, and rank and present recommendations. A variety of staff and colleagues peruse files and prepare summarizations and documentation, aided by databases, statistical tools, spreadsheets, etc. The associated tedium means that decisions, once made, are rarely withdrawn, even if facts, documented in updated databases, would indicate otherwise

Mediators are information processing modules that transform source data from distinct sources into actionable information. They automate a process within decision-making support that mimics activities carried out manually. The intent of the architectural model

is not to be exclusive and rigid. It is intended to provide a common framework under which many new technologies can be accommodated.

By automating the function within mediators such tasks are automated, and can be performed rapidly when the need arises, allowing the decision-makers to have access to the most recent information. The rules that drive mediation can often be obtained from experts that carried out such tasks in the past. Figure 35.5 conveys an impression of the current state of Mediation technology. As the systems become larger, responsiveness and efficiency must be addressed, but this issue is shared with the entire database community.

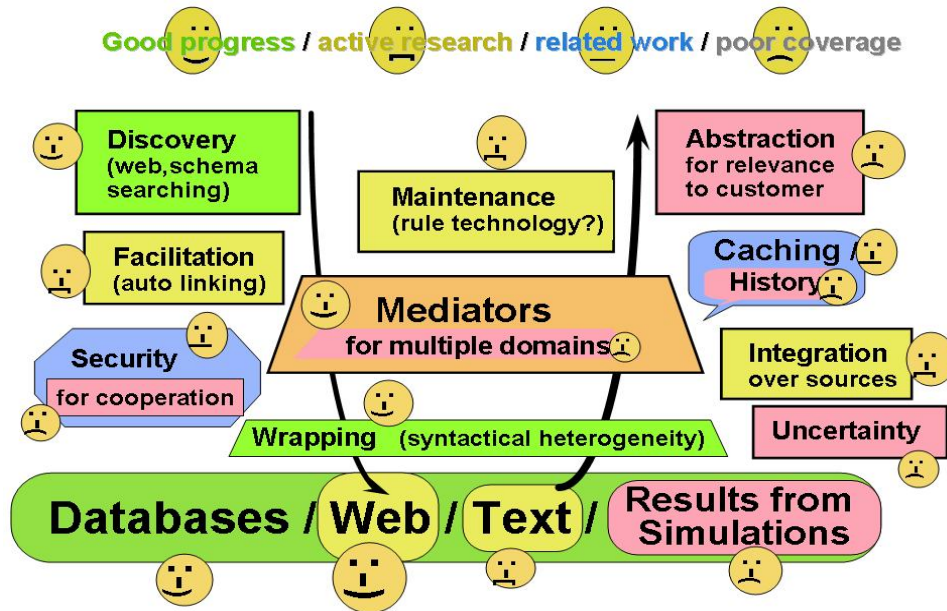


Figure 35.5: State of Mediation Technology

The resolution of semantics using terms from diverse sources is a task without end. That does not mean that no progress has or will be made. It is just that, as we learn to deal with one level of issues, further, finer shades of meaning become visible. The knowledge-based paradigms inherent in intelligent mediators indicate the critical role of artificial intelligence technology foreseen when implementing mediators. Mediators may be strengthened by having learning capability. Derived information may simply be stored in a mediator. Learning can also lead to new tactics of data acquisition and control of processing.

The technologies and issues presented in this chapter are and will be seen in many information systems, independent of the names and architectures used. In any case, a clear focus and organization can make such information systems effective, reliable, and maintainable.

## References

The references chosen are biased, because they are derived from work that we are familiar with. More references can be found in the cited works.

- [Atre:86] Shaku Atre: Information Center: Strategies and Case Studies}, Vol.~1; Atre Int. Consultants, Rye NY, 1986.
- [Berners-LeeHL:01] Tim Berners-Lee, Jim Hendler, and O. Lassila: "The Semantic Web"; *Scientific American*; 2001.
- [Bernstein 96] Bernstein, Philip A. "Middleware: A Model for Distributed Services"; *Communications of the ACM*, Vo. 39 No. 2, February 1996, pages 86-97.
- [BöhlenJ:07] Michael Böhlen and Christian Jensen: Temporal Databases; this volume.
- [Bush:45] Vannevar Bush: "As We May Think"; {\sl Atlantic Monthly}, Vol.176 No.1, 1945, pp.101--108.
- [CaceresFOV:06] Cesar Caceres, Alberto Fernandez, Sach Oosowski, and Matteo Vasirami: "Agent-based Semantic Service Discovery for Healthcare: an Organizational approach; IEEE Intelligent Systems, Vol.31 No.6, Nov.-Dec. 2006.
- [Cheeseman:85] Peter Cheeseman: "In Defense of Probability"; {\sl Proc. IJCAI}, Los Angeles, 1985, pp.1002--1009.
- [ChenM:89] M.C.~Chen and L.~McNamee: "A Data Model and Access Method for Summary Data Management"; {\sl IEEE Data Engineering Conf.~}5, Los Angeles, Feb.1989.
- [Connolly:97] Dan Connolly (ed.): XML: Principles, Tools, and Techniques; *O'Reilly*, 1997.
- [DaviesEa:03] John Davies, Dieter Fensel, and Frank van Harmelen: Towards the Semantic Web: Ontology-Driven Knowledge; Wiley 2003.
- [DeMichiel:89] Linda DeMichiel: "Performing Operations over Mismatched Domains"; {\sl IEEE Data Engineering Conference }5, Feb.1989; {\sl IEEE Transactions on Knowledge and Data Engineering}, Vol.1 No.4, Dec.~1989.
- [DeZegher:88] I.~DeZegher-Geets., A.G.~Freeman, M.G.~Walker, R.L.~Blum and G.~Wie\~der\~hold: "Summarization and Display of On-line Medical Records"; {\sl M.D. Computing}, Vol.5 no.3, March 1988, pp.38-46.
- [Fagin:99] Ronald Fagin: "Combining fuzzy information from multiple systems"; *Journal of Computer and System Sciences*, Vol.58 no.1, Feb.1999, Pages: 83 - 99. Academic Press.
- [FriederG:07] Ophir Frieder and David A. Grossman: "Information Retrieval Systems"; this volume.
- [Grust:07] Torsten Grust: XML Databases; this volume.
- [Guarino:98] N. Guarino: Formal Ontology in Information Systems; IOS Press, 1998.

- [HalevyIST:03] A.Y. Halevy, Z.G. Ives, D. Suci, and I. Tatarinov: "Schema mediation in peer data management systems" Proceedings. 19th International Conference on Data Engineering, March 2003, pages 505- 516.
- [HalpernK:04] J. Y. Halpern and D. Koller (2004). "Representation dependence in probabilistic inference"; *Journal of Artificial Intelligence Research*, Vol. 21, pages 319-356.
- [HammerGNYBV:97] Joachim Hammer, Héctor García-Molina, Svetlozar Nestorov, Ramana Yerneni, Marcus Breunig, and Vasilis Vassalos: "Template-based wrappers in the TSIMMIS system", Proceedings ACM SIGMOD International Conference on Management of Data, 1997, pages 532 - 535, ACM Press.
- [Hayes-Roth:84] Frederick Hayes-Roth: "The Knowledge-based Expert System, A tutorial"; *IEEE Computer*, Sep.1984, pp.11-28.
- [Holsapple:04] C. Holsapple (ed.): *Handbook on Knowledge Management*; International Handbooks on Information Systems, Springer Verlag, 2004.
- [IpeirotisAJG:06] Panagiotis G. Ipeirotis, Eugene Agichtein, Pranay Jain, and Luis Gravano: "To Search or to Crawl? Towards a Query Optimizer for Text-Centric Tasks"; *ACM SIGMOD*, 2006.
- [Jackson:99] Peter Jackson: *Introduction to Expert Systems*, 3rd edition; Addison Wesley Longman, 1999
- [Jain:91] Raj Jain: *The Art of Computer Systems Performance Analysis*; Wiley 1991.
- [Jarke:07] Mathias Jarke: Data Warehousing; this volume.
- [Keller:07] Arthur Keller: Object-Oriented Databases; this volume.
- [Kim:95] Won Kim (ed): *Modern Database Systems: the Object Model, Interoperability and Beyond*; ACM press, Addison Wesley, 1995.
- [KnoblockMAAMPT:01] C. Knoblock, S. Minton, J.L. Ambite, N. Ashish, I. Muslea, A. Philpot, and S. Tejada: "The ARIADNE Approach to Web-based Information Integration"; *International Journal of Cooperative Information Systems*, 2001, Vol. 10, no.1-2, pages145-169.
- [KollerH:92] D. Koller and J. Y. Halpern (1992). "A logic for approximate reasoning." *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pp. 153-164.
- [LinOA:04] Zhen Lin, Art B. Owen, Russ B. Altman: " Genomic Research and Human Subject Privacy"; *Science*, Vol. 305 No. 5681, p. 183, 9 July 2004.
- [MalufW:97] David A. Maluf and Wiederhold Gio, "Abstraction of Representation for Interoperation," *Foundations of Intelligent Systems*, Zbigniew Ras and Andrzej Skowron Eds., Lecture Notes in Artificial Intelligence, Springer Verlag, Vol. 1315, 1997.
- [Miller:56] George Miller: "The Magical Number Seven  $\pm$  Two"; *Psych.Review*, Vol.68, 1956, pp.81-97.

- [MitraWK:00] Prasenjit Mitra, Gio Wiederhold, and Martin Kersten: "A Graph-Oriented Model for Articulation of Ontology Interdependencies"; in Zaniolo et al. (eds): Extending DataBase Technologies, Springer Verlag LNCS 1777, March 2000.
- [MitraW:04] Prasenjit Mitra and Gio Wiederhold: "An Ontology-Composition Algebra"; in S.Staab, R.Studer (eds.): Handbook on Ontologies, Springer Series: International Handbooks on Information Systems, 2004, pages 93-113.
- [MusleaMK:99] Ion Muslea, Steve Minton, and Craig Knoblock: "A hierarchical approach to wrapper induction"; International Conference on Autonomous Agents, Proceedings of the Third Annual Conference on Autonomous Agents ACM Press, pages 190 - 197, 1999
- [MutsuzakiEa:07] Michi Mutsuzaki, Martin Theobald, Ander de Keijzer,; Jennifer Widom, Parag Agrawal, Omar Benjelloun, Anish Das Sarma, Murthy Raghotham, and Tomoe Sugihara: "Trio-One: Layering Uncertainty and Lineage on a Conventional DBMS"; Proc. of CIDR conference, VLDB Foundation, Jan. 2007.
- [NoyM:00] Natasha F. Noy and Mark A. Musen: PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment; Proceedings of the National Conference on Artificial intelligence, AAAI, 2000.
- [Olken:86] F.~Olken and D.~Rotem: "Simple Random Sampling from Relational Databases"; {\sl VLDB }12, Kyoto, Aug.1986.
- [Orman:88] Levent Orman: "Functional Development of Database Applications"; {\sl IEEE Trans. Software Eng.}, Vol.14 No.9, Sep.1988, pp.1280--1292. = Since observed events are recorded as opposed to the derived state of the environment, the need for triggers is minimized.
- [Risch:89] Tore Risch: "Monitoring Database Objects"; {\sl Proc. VLDB } 15, Amsterdam, Aug.~1989, Morgan Kaufmann Pubs.
- [Rowe:83] Neil~Rowe: "An Expert System for Statistical Estimates on Databases"; {\sl Proc. AAAI}, Stanford, Mar.1983.
- [Shannon:48] C.E.~Shannon and W.~Weaver: {\sl *The Mathematical Theory of Computation*};1948, reprinted by The Un.Illinois Press, 1962.
- [StaabSSS:01] Steffen Staab, H.-P. Schnurr, Rudi Studer, Y. Sure: "Knowledge Processes and Ontologies"; *IEEE Intelligent Systems*, Vol. 16 No. ), January/February 2001.
- [Stonebraker:86] M.~Stonebraker and L.A.~Rowe: "The Design of POSTGRES"; *Proc. ACM SIGMOD'86*, May.1986, pp.340--355.
- [Ullman:00] Jeffrey D. Ullman: "Information integration using logical views" Theoretical Computer Science, Issue 239, 2000, pages 189-210, Elsevier Publishers.
- [Waldrop:84] M.Mitchell~Waldrop: "The Intelligence of Organizations"; {\sl *Science*}, Vol.225 No.4667, Sep.1984, pp.1136--1137.
- [Wolfson:07] Ouri Wolfson: Spatio-Temporal Databases; this volume.

- [Wiederhold:92] Gio Wiederhold: "Mediators in the architecture of future information systems"; *IEEE Computer*, Vol. 25 no.3, 1992, pages 38-49.
- [WiederholdJL:93] Gio Wiederhold, Sushil Jajodia, and Witold Litwin: "Integrating Temporal Data in a Heterogenous Environment"; in Tansel, Clifford, Gadia, Jajodia, Segiv, Snodgrass: *Temporal Databases, Theory, Design and Implementation*; Benjamin Cummins Publishing, 1993, pages 563-579.
- [Wiederhold:96], Gio Wiederhold (editor): [\*Intelligent Integration of Information\*](#); Kluwer Academic Publishers, Boston MA, July 1996.
- [WiederholdG:97] Gio Wiederhold and Michael Genesereth: "The Conceptual Basis for Mediation Services"; *IEEE Expert*, Vol.12 No.5, Sep.-Oct. 1997, pages 38-47.
- [Wiederhold:99] Gio Wiederhold: "Mediation to deal with Heterogeneous Data Sources"; in Vckovski, Brassel, and Schek: *Interoperating Geographic Information Systems*, Springer LNCS 1580, pages 1-16.
- [Wiederhold:00C] Gio Wiederhold : "Future Needs in Integration of Information"; *International Journal of Cooperative Information Systems*; special issue on Intelligent Integration of Information, Vol. 9, no.4; World Scientific pubs., 2000, pages 449-472.
- [Wiederhold:00I], Gio Wiederhold: "Information Systems that Really Support Decision-making"; *Journal of Intelligent Information Systems*; Vol.14, Kluwer, March 2000, pages 85-94.
- [Wiederhold:01S] Wiederhold, Gio: "Collaboration Requirements: A Point of Failure in Protecting Information"; *IEEE Transactions on Systems, Man and Cybernetics*, Vol.31 No.4, July 2001, pages 336-342.
- [Wiederhold:02D] Gio Wiederhold: "Information Systems that also Project into the Future"; in Subhash Bhalla, ed.: *Databases in Networked Information Systems*, Dec. 2002, Springer LNCS 2544, pages 1-14.
- [Wiederhold:02E] Gio Wiederhold: "Obtaining Precision when Integrating Information"; in J.Filipe, Sharp, B. and Miranda, P. (Eds.), *Enterprise Information Systems III*, Kluwer Academic Publishers, 2002.

===== o ===== o =====