Understanding Structural and Semantic Heterogeneity in the Context of Database Schema Integration

David George, Department of Computing, University of Central Lancashire, Preston UK, DGeorge@uclan.ac.uk

Abstract

Over the last 25 years data integration has been a key issue in achieving systems interoperability, between heterogeneous data storage and management systems, because of the existence of system, schema, and semantic heterogeneity. This paper starts by examining the background to data integration by discussing why integration has remained such an issue, in terms of current rationale and drivers for data integration. Previous integration strategies are briefly discussed and these encompass three generations of methodologies. They address multiple schema integrations, using federated databases and mediator/wrapper systems and, more recently, the trend towards information brokering using ontology-based approaches. However, the integration process primarily requires understanding of the characteristics of heterogeneity, so that database schemas can be examined to identify semantically related conflicts that may then be transformed, or mediated, to deliver interoperable data stores. Therefore, the more detailed consideration in this paper is towards an understanding of both the causes and examples of structural (schema), syntactic (format) and semantic (meaning) heterogeneity, created during the data model and schema design process. Finally, the Semantic Web is introduced because associated technologies may provide the language specification to underpin third generation approaches of ontology development.

1 Introduction

Organisations manage the complexity of society by adapting to the needs of customers and other organisations and evolving new corporate strategies that drive organisational changes through business reorganisation and merger. New information systems monitor performance, identify new business opportunities, and meet legal requirements (Rob and Coronel, 2002). However, developing organisations are also characterised by retention of legacy systems (Stonebraker et al., 1993), resulting in the need to integrate legacy and new database management systems (DBMSs). This was demonstrated by the problems at US West (Drew et al., 1993).

2 The Data Integration Challenge

Database (DB) technology was created to resolve the problems of data fragmentation and redundancy evident in organisations (Rob and Coronel, 2002, Howe, 2003). The last 30 years have witnessed two paradigms in the data integration challenge; the development of the entity-relationship (E-R) and object-oriented (O-O) models (Chen, 1976, Kim, 1991). Whilst DB technology has had a significant impact on this problem in the past, exponential growth in diverse information accessed on the Web has made information retrieval increasingly complex, with billions of documents being accessed by over 300 million users (Patel-Schneider and Fensel, 2002). The combination of structured DB sources, semi-structured and unstructured Web data means that systems interoperability and online-data integration represent some of the most significant challenges facing the information technology (IT) community in the last 25 years (Sheth, 1998), with the cost of data integration and improving data quality estimated at $1bn a year (Brodie, 2003).

Global organisations, with geographically isolated operations, share access to distributed and heterogeneous data sources. Further, organisations now need to reuse and analyse shared data to acquire information and knowledge that can underpin business intelligence programmes (Hill, 2004). They rely on decision-making support, full customer product

holding information at point-of-sale, and customer segmentation analysis, for deploying marketing and sales initiatives.

3 Integration Architectures

The problem of DB integration manifested itself as a result of the development of multidatabase systems (Batini et al., 1986). These were classified as either *homogeneous* distributed systems, dealing with local DBs having the same data model and languages, or *heterogeneous* systems, containing diverse DBMSs, languages and data models. This has resulted in three generations of evolution in levels of information systems interoperability: the period to the mid-eighties, the period to the mid-nineties, and the mid-nineties to date (Sheth, 1998).

3.1 Integration Architectures – Generations 1 and 2

In the first generation, companies were characterised by having large volumes of data in different departments, yet needing to exchange and share data between departments. Multidatabase and *federated DB systems* (FDBMSs) primarily focused on systems interoperability of DBMSs and later on schema integration through global and federated schemas (Sheth and Larson, 1990, Drew et al., 1993, Bright, 1994). The emphasis was on system and data management, as opposed to information or knowledge management, involving small-scale distributions of mainly relational and entity-relationship DBs. These systems provided limited scalability.

With the expansion of the Internet and Web, second-generation integration saw increasing development of *federated information systems* that addressed both structured DBs and a wider range of data sources including text repositories, semi-structured data, and digital media such as images and video. These systems included mediator/wrapper architectures that generate a mediated schema as a *homogeneous* and *virtual* information source, without integrating the data resources, and other online information systems making more extensive use of metadata (Wiederhold, 1992, Levy et al., 1996, Garcia-Molina et al., 1997, Bertino et al., 2001). Metadata encompassed a variety of forms beyond simply schema, including DB descriptions, content descriptions of images and audio, and HTML/SGML document type definitions. Interoperability was focused more on structure (data schema) and syntax (data types) than systems, and wider-scale network distributions that showed increasing evidence of object-oriented DBMSs.

3.2 Integration Approaches - Generation 3

Progress in global interconnectivity has provided access to billions of information resources often relying on simple keyword searches via search engines. However, as keyword searches deliver only limited precision in identifying relevant information, the main challenge has progressed to a semantic level, with users now requiring machine support to also understand the contexts of such diverse resources. Progression from storing data to managing information, and knowledge acquisition, has made the need for semantic interoperability more apparent.

Enterprise-wide and global challenges require the content and representation of information to be more closely related to domain specific concepts, such as by using shared ontologies (Gruber, 1993, Guarino, 1998). The predominant architectures are multi-modal information brokering systems (Ouksel and Sheth, 1999, Bergamaschi et al., 1999), with semantics described by potentially multiple ontologies (de Bruijn, 2003) and artificial intelligence (AI) for information queries of global information systems (GIS). Ontologies, in a computing context, are a formalised vocabulary of concepts, their relationships and explicit assumptions

of a subject domain, and represent an agreed "universe of discourse" to relate information structures.

3.3 Integration Summary

Technologies have progressed from traditional DB information architectures to information brokering and retrieval, and knowledge-based systems using AI and digital media. Achieving departmental interoperability by reconciliation of existing disparate DBs "after the event", has progressed to encompassing enterprise-wide, inter-enterprise and global interoperability using formalised ontology structures as anchors to contextualise structured and semi-structured data. Shared ontologies offer a solution to enable participating information systems to be described, and similarities between related objects to be determined (Fensel, 2001). Nevertheless, a focus on the key elements in integrating DB systems is relevant at this stage.

4 Dimensions of Database Integration

Interoperability through integration of heterogeneous DB systems involves the dimensions of system, schema, and semantic integration. Researchers and developers have sought to resolve integration of such heterogeneity for many years (Fang et al., 1994, Bukhres et al., 1996, Ouksel and Sheth, 1999, Sheth, 1998).

4.1 System Integration

Although system integration is not part of this review, it facilitates data access from multiple DBs and refers to platform heterogeneity in operating system and hardware, and information system heterogeneity, such as DBMSs and digital media repository systems (Sheth, 1998)

4.2 Schema and Semantic Integration

Schema integration is the process of merging autonomously developed DB schema into a unified, global schema to provide transparency through a unified view – see Figure 1.

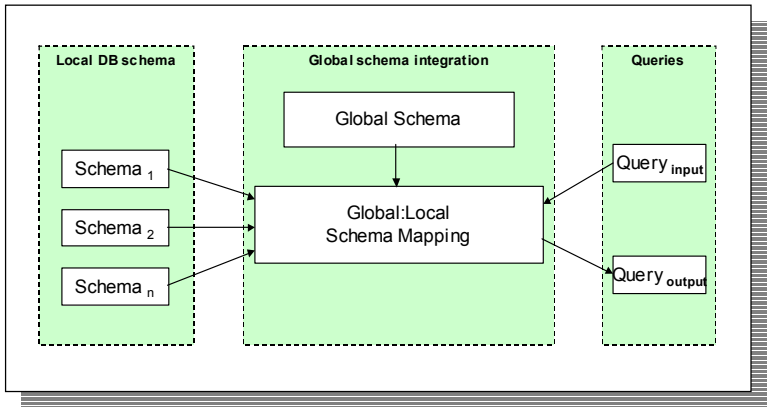

Fig. 1. Schema Integration

In federated systems, sharing is made explicit by autonomous local DBMSs allowing export schemas to serve as the shareable components in a federation of schemas (Batini et al., 1986). Equally, schema integration may occur in single large systems, where schemas (or views) may have been designed for each user group; a view integration approach, of designing individual schemas that are subsequently merged, may then be used (Elmasri and Navathe, 2000). Schema integration has been variously described as a 3, 4 or 5-step process (Batini et al., 1986, Sheth and Larson, 1990, Parent and Spaccapietra, 1998) and involves the tasks of *pre-integration* (schema translation into common data model form), *comparison* (process of semantic conflict identification), *conformance* (making conflicts compatible for merging by

similar representation), and *merging* (integrating schemas) including *restructuring* (refining schema).

Semantic integration, implicit within schema integration, resolves differences in conceptual representation of data by determining equivalence between schema constructs and removing ambiguity among component DBs (Garcia-Solaco et al., 1996, Kashyap and Sheth, 1996).

5 Causes of Schema and Semantic Heterogeneity

Two issues play a significant role in creating disparities between DBMSs, namely organisational islands of development and differing designer influences in the developer process.

5.1 Development Autonomy

Islands of development occur where organisations have evolved as collections of distinct, autonomous departments with disconnected systems; each pursuing its own IT infrastructure (Lamb and Davidson, 2000). Alternatively, a DB structure may be simply too complex to be modelled by one designer. An example of the former was experienced in financial services, where mortgage, savings, and insurance departments were historically developed autonomously and specialised, heterogeneous systems were often bought-in to support new business activities.

5.2 Design Autonomy

Design autonomy can be reflected in differing designer influence and perception of the universe of discourse, data model representation (model and query language), naming conventions, semantic interpretation of data, and constraints applied (Batini et al., 1986, Sheth and Larson, 1990, Bukhres et al., 1996). Design autonomy produces differing perspectives, equivalence (but not identical) and incompatible design specifications.

Different perspectives can reflect different structures, such as one schema showing a relationship S1(Employee:Dept) versus another showing S2(Employee:Project:Dept), or a name inconsistency between related entities or attributes. *Equivalence* among model constructs exists when different constructs are used to model the concept equivalently e.g. where entities in one schema are modelled as attributes in another or where there are generalisation or specialisation differences e.g. in super-class:subclass structures. *Incompatible design specifications* are caused by specification of different data types, cardinality or referential integrity.

5.3 Modelling the Real World in Database Design

Semantic heterogeneities represent differences in the real world interpretation of context, meaning, and use of data and occur during the designer's task of translating conceptualisations of the real world into DB world representations - see Figure 2.

They reflect data model, schema construct, and data inconsistencies in the conceptual and database worlds (Kim et al., 1993, Hammer and McLeod, 1993, Kashyap and Sheth, 1996, Garcia-Solaco et al., 1996). Where two objects represent the same *concept* (of the entity or object) there may be a semantic relationship, or equivalence, but if the *contexts* (universe of discourse) differ, such as considering employees in two separate companies, then different extensions will result i.e. different instances of employee. Conversely, where extensions are the same in two entities they may be semantically unrelated e.g. two identical groups of people but one group happens to represent an operational department and one a project team. Semantic understanding is based on the relationship between concept and context. Identification of semantic heterogeneity requires consideration of both issues.

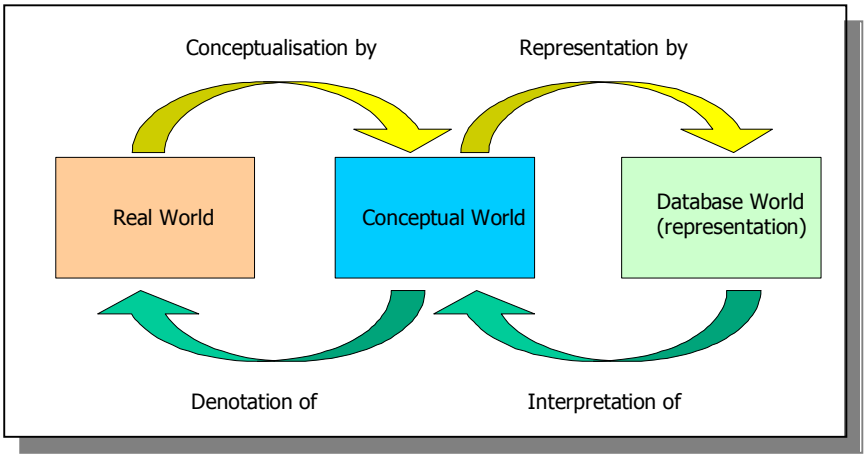

Fig. 2. Relationship between Real, Conceptual and Database Worlds

6 Heterogeneities resulting from Autonomy

The following two sections examine heterogeneity when respective concepts are related but their representations differ (Pitoura et al., 1995). The first section discusses taxonomies of types of heterogeneity and provides examples of such inconsistencies. The second section examines measurement of semantic relationships, to define the degree of semantic similarities between objects. Both sections provide an understanding of conflict identification and reconciliation.

6.1 Classifications of Schema and Semantic Heterogeneity

Various classifications of heterogeneities have been suggested in papers related to data integration, without necessarily providing full classifications. In an analysis of schema integration methodologies (Batini et al., 1986), structural and semantic diversity categories were specified as those involving naming conflicts and those involving structural conflicts.

Naming conflicts occur when different terminology is used across organisations. Differences in entity or attribute naming are classified as either homonyms (differing concepts but having same name) or synonyms (same concepts but having different names).

Structural conflicts occur when a different choice of modelling construct or integrity constraints is adopted. Four categories of conflict were identified.

1. *Type conflicts* relate to modelling constructs, such as when concepts are represented as an entity in one schema compared to an attribute in another. This conflict is demonstrated by the specification of Publisher as an entity versus attribute in Figure 3.

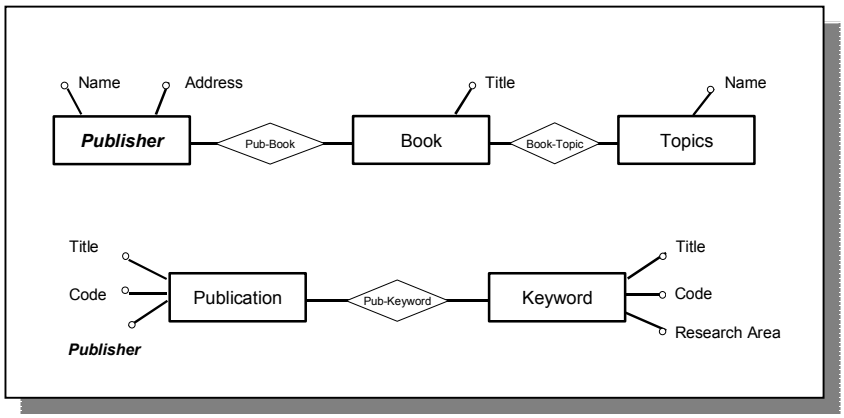

Fig. 3. Publisher entity versus attribute conflict

A further example can be demonstrated where equivalent constructs are represented in a generalisation hierarchy, for example where one schema contains an entity or class Student, with specialisation types U-Graduate and Graduate, whereas another represents it as an entity or class generalisation Student as in Figure 4.

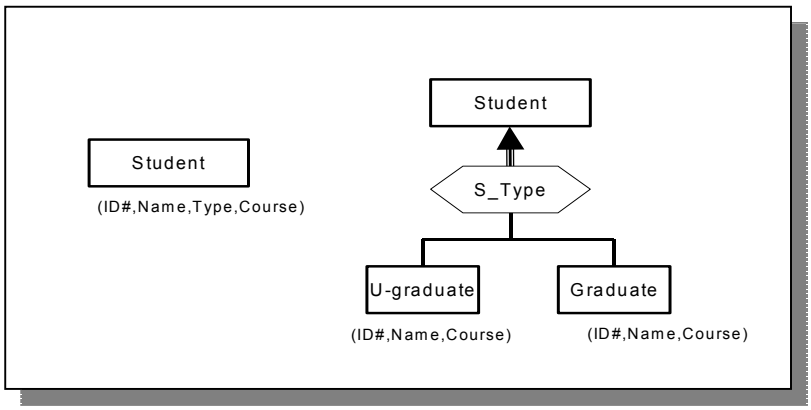

Fig. 4. Student generalisation conflict

2. *Dependency conflicts* can be shown in a relationship between entities Employee and Employment i.e. in one schema they could be represented as a 1:1 relationship (showing current role) compared to a 1:M relationship (current and previous positions).
3. *Key conflicts* may apply in the design specification of a primary key for Employee. It may be ss# in one schema and empID in another.
4. *Behavioural conflicts* may show differences in insert and delete policies, such as where one schema allows a department instance to exist without employees whereas an employee deletion could result in a department deletion in another schema.

In a survey of heterogeneous systems of FDBMSs (Sheth and Larson, 1990), heterogeneity was highlighted in two areas: *database management systems* and *heterogeneity of semantics*.

Database management systems involved structural differences (E-R versus Network model, entity versus attribute, and systems differences in how they supported generalisation and inheritance), constraint differences (e.g. specification of referential integrity), and query language differences (e.g. SQL versus OQL). Heterogeneity of semantics included differences in definition (e.g. attribute flightFare includes airport tax compared to flightCost excludes tax) and precision (e.g. Student grades stored on a scale of 1:100 versus A-E).

A further study of heterogeneity in federated systems (Hammer and McLeod, 1993) defined the categories as differences in: *metadata specification* of the conceptual schema (conflicts in structure of relationships), *object comparability* (e.g. in naming: synonyms and homonyms), and *low-level data format* (in atomic data values e.g. units of measure). Numerous other publications address these issues to differing degrees (Pitoura et al., 1995, Bukhres et al., 1996, Song et al., 1996, Colomb, 1997, Tseng et al., 1998, Parent and Spaccapietra, 1998). However, several researchers have constructed wide-ranging classifications of schema conflict (Sheth and Kashyap, 1992, Kim et al., 1993, Garcia-Solaco et al., 1996).

The work by Kim (1993) examined structural conflicts and data conflicts based on integrations of differing models. It adopted a convention to enable comparison between entity-relationship (E-R) and object-oriented (O-O) schemas; the term *entity* corresponding to table in E-R and class in O-O, with *attribute* corresponding to column in E-R and attribute in O-O. The classification identifies two key causes of conflict: where component schemas use different *structures* to represent the same information and, where different *specifications* are

used for related or similar structures, e.g. names or domains. A summary of the classification follows.

Conflicts between Entities

a. One-to-One Entity Conflicts.

This includes *entity name conflicts*, i.e. different names for equivalent entities (synonym) or same name for different entities (homonym), and *entity structure conflicts*, i.e. missing attributes (differences in number of attributes) or missing but implicit attributes (attributes can be deduced).

Other conflicts are *entity constraint conflicts* in the specification of keys and check conditions and *entity inclusion conflicts* in generalisation and associated inheritance.

b. Many-to-Many Entity Conflicts

Conflicts are as in *(a)* but occurring when schemas use different numbers, or subcategories, of entities to represent the same information.

Conflicts between Attributes

a. One-to-One Attribute Conflicts.

This includes *attribute name conflicts*, i.e. different names for equivalent attributes or same name for different attributes and *attribute constraint conflicts*, i.e. integrity constraints, data type (difference between data primitives and user-derived types), and composition (differences in aggregation). Other conflicts are *default value conflicts*, *attribute inclusion conflicts*, i.e. attribute generalisation and inheritance issues and *method conflicts*, i.e. missing or differing methods.

b. Many-to-Many Attribute Conflicts

Conflicts are as in *(a)* but occurring when schemas use different numbers, or subcategories, of attributes to represent the same information.

Conflicts between Entities and Attributes

These represent a combination of the entity attribute classifications, such as when one schema uses an entity and another uses an attribute to represent the same information.

Data Conflicts

These encompass *different representation for equivalent data*, i.e. different expressions (differing scalar values), different units, or different precisions (scale of precision). Other conflicts are *wrong data*, i.e. incorrect entry or obsolete data, *incomplete data* – missing data and *noisy data*, i.e. unexpected errors.

A comparison of this taxonomy, with that of Sheth and Kasyap (1996), shows a significant similarity in conflict classifications. However, Kim et al (1993) also identified potential object *method* conflicts, although argued that they could be treated in the same way as attributes i.e. by name conflicts, missing, and data types. Sheth and Kasyap (1996), in similarly identifying entity versus attribute conflicts (terming them schematic discrepancies), also identified data value versus attribute and data value versus entity conflicts. However, both classifications appear to be subsumed in a comprehensive and revised taxonomy proposed by Garcia-Solaco et al (1996). The following analysis seeks to provide a concise explanation of the principles of heterogeneities grouped on the basis of object classes, class structures, and object instances.

Semantic Heterogeneities between Object Classes

1. Differences in *extensions*, where extensions of a class are the set of objects (instances) that are members of the class at a point in time:
    a. differences in *characterisation* of object set of a class e.g. class Colours(ROYGBIV) vs. (ROYGBIV + Pink) vs. characterisation by colour wavelength intervals. This demonstrates different cardinalities, i.e. 7:8, and different aggregation (see class structures item 2) i.e. wavelength intervals may differ or may embrace two colours e.g. what corresponds to Pink? (see also *Precision* differences 4.b.vi).
    b. object *inclusion* differences e.g. concepts of DB1 "employees" and DB2 "emp" may agree but context may differ as it may not be same company and employees. Also customers of one superstore may or may not be same as customers of another.
    c. object *multiplicity* differences e.g. DB1 Employees has one class with all attributes of employee each represented by one object, whereas DB2 Employees contains sub classes "sales" etc. and employee is represented as employee and sales objects.
2. Differences in *names*, involving class and attribute/method *synonymy* of names (same concept different names, e.g. Customer vs. Client) and *homonymy or polysemy* (same names different concepts e.g. attribute *market* relates to product vs. customer).
3. Differences in *attributes and methods* demonstrated by:
    a. *presence/absence* differences between attributes specified - such as two addresses may contain the same attributes, except one contains an additional email attribute.
    b. *temporal* differences, for example current employee salary vs. past salary history.
    c. -*arity* differences, such as where a stock price is specified as a £value *attribute* vs. a £value *method,* generated form the product of quantity x price.
    d. *constraints* differences between single attributes, including *multivalue* differences (single vs. multiple cardinality e.g. employee job v jobs, or different minimum and maximums allowed), *nulls* differences where nulls are allowed vs. no-nulls, and *uniqueness* differences in attributes e.g. using primary key vs. non-primary key.
    e. *default value* differences involving differing values or values vs. no values.
4. Differences in the *domains* of attributes/*results* of the methods include:
    a. *semantic domain* differences - conceptualisation of the values and operations used:
        i. differences in *object identification* includes: *system/application identifier* differences, i.e. system-generated OIDs vs. application-created keys, differences in *keys*, i.e. use of different encoding schemes e.g. emp_Names vs. emp_IDs vs. emp_SSNs, and differences at the "class" level e.g. Customer.Code vs. Client.Number even though it may be the same key at instance level.
        ii. *numerical* v *non-numerical*, such as demonstrated in grades A-C vs. 1-5.
        iii. *dimension, units of measure and scale* differences. *Dimension* involves volume, weight, price, size, *units of measure* can be reflected in dimensions of miles vs. kilometres, Celsius vs. Fahrenheit, and *scale* based on the selected dimension and applied measure e.g. £000s vs. $millions.

iv. *monetary* differences showing conflicting usage of $, £, yen, euro, and *date differences* e.g. dd.mm.yy vs. mm.yyyy, long vs. short dates, and *default value* differences.

b. *syntactic domain*: are *type* differences, regardless of the semantic domain conflict:

i. *system OID* differences, such as OID are unknown in OODB1 vs. OODB2.

ii. *length* differences, such as fixed vs. variable length strings, or where both are fixed but at a different fixed length.

iii. *character vs. numerical* differences, e.g. empID 4536 specified as integer vs. string, and *numerical type* differences involving integer vs. fixed or float.

iv. *base* differences, for example 2, 8, 10, 16 and *precision* differences e.g. employing fixed vs. floating point, or 1-100 vs. A-F.

v. *integer* differences – long vs. short and *integral length* - decimal places.

vi. *special case*, such as date expressions e.g. dd/mm/yy vs. dd-mm-yyyy.

5. *Constraint* differences that relate to conflict in *conditions between* several attributes e.g. one schema may specify if female and married then maiden name not null.

Semantic Heterogeneities between Class Structures

1. *Generalisation*/S*pecialisation* inconsistencies reflect heterogeneities between "*classifications*" in the super-class/sub-class dimension:

a. *criteria based* differences, such as employees specialised as male and female groups vs. occupation groupings shows specialisation difference - Figure 5.

b. *degree and characterisation based* differences, such as specialisation of customers by differing age groupings or by life-stage types - Figure 5.

c. *specialisation kind* differences, such as specialisation of employee by job types vs. by the job listed for each employee.

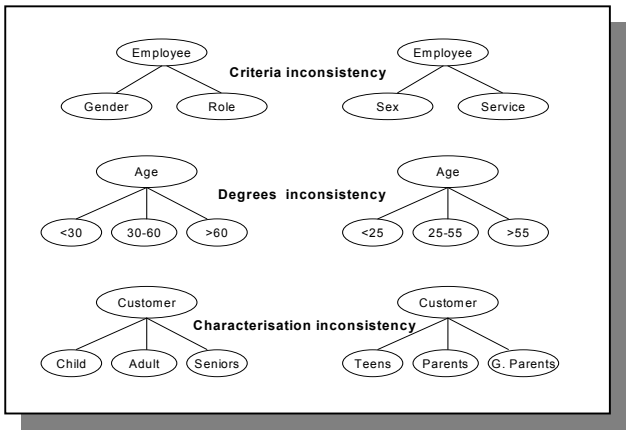

Fig. 5. Examples of criteria, degree and characterisation differences

d. *constraint* differences, for example when object removed from a class it is also removed from all sub-classes e.g. where multiple super-classes of object.

2. The *Aggregation*/*Decomposition* dimension reflects aggregation conflicts involving three types of aggregation: (a) *simple aggregation*, where the object class is an aggregation of its attributes e.g. attributes that have as their domains other classes, (b) *composition*, where an object class represents an aggregation of attribute classes e.g. WorkProgram can not exist without both of Employees and Projects, and (c)

collection, where an object class is created as a collection of objects of one class. These can be represented by:

a. *simple* classes vs. *aggregated* classes e.g. objects of class Yellow vs. Yellow composed of 3 objects of class PrimaryColours being Red, Blue and Green.

b. inconsistencies in the *kind of aggregation* e.g. class Department collection of employee *attributes* vs. simple aggregation using a *multi-valued* employee attribute.

c. inconsistencies within the *aggregated* classes themselves, by *aggregation* e.g. WorkProgram(employee, project) vs. (employee, subproject), by *specialisation* e.g. CarType(carMake, carDesign) vs. FamilyType(carMake, saloonSize), and by *composition* e.g. Person(address, tel.) vs. Person(street, city, county, tel.).

d. inconsistencies in the *sub-kind of the collection* e.g. Dept(employees), where an employee works in only one department vs. works in more than one department.

e. inconsistencies in the *component class of the collection* e.g. Employee(Dept) vs. Employee(Division(Dept)).

f. *aggregation delete effect*, where there is differing treatment of deletions e.g. in Car(engine, air-conditioning), engine object attribute deletion may be rejected.

3. *Schematic discrepancies* exist where *data*, e.g. value "Father", in DB1 is represented as *metadata* in DB2 (schema attribute) and DB3 (schema class Fathers) as in Figure 6. Parent-Type attribute in DB1 has a value of either Father or Mother. Therefore, in this comparison, Parents would be identified by two objects in DB1, by one object in DB2, and by two objects in DB3. These can demonstrate *data-metadata* conflicts, such as in the specialisation and composition examples in a and b:

   a. *specialisation* discrepancy i.e. DB1:Parenthoods vs. DB3:Fathers/Mothers and shows a value-entity difference.

   b. *composition* discrepancy – i.e. DB1 would contain two Parenthood objects compared to one Parenthood object in DB2. This shows a value-attribute difference.

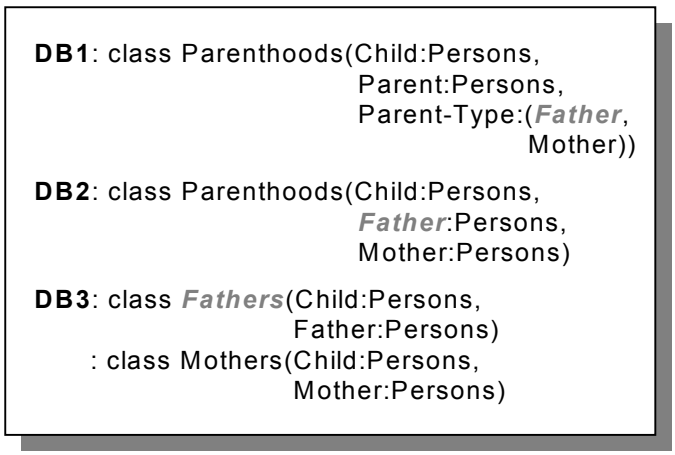

Fig. 6. Schematic discrepancies

   c. *composition and specialisation* discrepancy – i.e. DB2: Parenthoods via one object vs. DB3: Fathers/Mothers via two objects. This shows an attribute-entity difference and therefore no data-metadata conflict.

A further conflict can relate to *collection and specialisation* discrepancies e.g. Departments having an aggregate collection of Employees vs. Employees with an aggregate collection of Departments (a subclass in this case).

Semantic Heterogeneities between Object Instances

Four types of conflict exist: *presence/absence* discrepancies, i.e. there is no matching object instance between two DBs, *multi-valued* attribute discrepancies involving different numbers of values, *nulls/nonulls* discrepancies in object instances, and attribute *value* discrepancies, such as where vehicle A234 PNE colour is stated as blue in one schema but red in another.

This classification is a detailed taxonomy of schema and semantic heterogeneity. However, further types of linguistic conflicts between schema terms have been identified. These relate to *hypernyms*, which apply broader meaning or generalisations e.g. for use in global schema query terms, and *hyponyms* (opposite of hypernyms), which are more specific e.g. use of is-a, part-of, member-of, form-of etc. These are used to build hierarchical structures that describe information contained in component DBs and are relevant when developing a common data model or a common thesaurus (Bright, 1994, Bergamaschi et al., 1999, Palopoli et al., 2002).

6.2 Measure and Detection of Semantic Relationships

At this stage, it is relevant to appreciate some early descriptions of measures of semantic relationships, designed to reinforced understanding and identification of schematic differences (Batini et al., 1986, Hammer and McLeod, 1993). These measures were identified as:

1. *identical* – where the same perceptions and modelling constructs are applied with no evidence of incoherence. Although, Hammer and Mcleod (1993) considered basic name conflict acceptable e.g. schemaA.Reservations is coherent with schemaB.Bookings.
2. *equivalent* – where constructs are not identical but concepts are the same and coherent e.g. in schema1.Airlines versus schema2.EuropeanAirlines, both model the same information but location is not implied in schema1.
3. *compatible* – where representation is neither identical nor equivalent but both model the same information e.g. schema1.PrivateAccommodations versus schema2.BeachResorts are not contradictory but the semantics are not the same.
4. *incompatible* – where the design specification is contradictory because of specification incoherence e.g. for invoices referencing orders, one schema may specify a cardinality of 1:1 relationship, whereas it may be 1:M in another schema.

For schema integrations, schematic conflicts are only of interest when there is a semantic relationship. Measures 2 to 4 represent meaningful conflicts that are related in some way, and are therefore likely to demand solutions to achieve schema integration.

Other classifications or taxonomies have been developed to characterise the degree of semantic similarities between objects. The Semantic Proximity Model (SemPro) was developed to model uncertainty, inconsistency and incompleteness in information matching (Sheth and Kashyap, 1992, Kashyap and Sheth, 1996). SemPro provides a qualitative measure between two objects compared, defined by a context representation or semantic proximity descriptor that captures the *context* of comparison of the objects and the *abstraction*/mapping relating the object domains. The detail of this approach is beyond the scope of this paper but a brief explanation can be provided using the following definition syntax and subsequent example:

$semPro(O_1, O_2) = <Context, Abstraction, (D_1, D_2), (S_1, S_2)>$

Where D = domain of O (attribute) and S = state of O (extensions of the objects recorded at a particular time). Thus, using a simple example of two entities depicted in different schemas:

$Employee_1$(ID#, homePhone, officePhone)
$Employee_2$(ID#, phone)

The semantic similarity would be represented by:

$SemPro(Employee_1, Employee_2) = <ALL,[M_{ID},M_1],([D_{1,ID\#}, D_{homePhone}, D_{officePhone}],[D_{2,ID\#}, D_{phone}]), (S_1, S_2)>$

Where M_{ID} demonstrates that a 1:1 mapping exists between keys $D_{1,ID\#}$ and $D_{2,\ ID\#}$ and where M_1 represents a possible total/partial mapping between $(D_{homePhone} \cup D_{officePhone})$ and D_{phone}.

The SemPro approach provides a means to qualitatively measure conflicts between two objects, by defining five *degrees* of semantic relationship. The first is termed *semantic equivalence*, and applies where two objects represent the same real world entity or concept i.e. there is a total 1:1 mapping between the domains of the objects in any context e.g. entity Client versus entity Customer, or where the same vehicle model price is stated in £ but in $ elsewhere. The second is *semantic relationship* and represents a weaker type of semantic similarity, where there is a partial M:1 mapping or generalisation/aggregation abstraction between the object domains e.g. given $Object_1$ it is possible to identify $Object_2$ but not vice versa e.g. entity FootballTeam(name, ground) versus FootballPlayer(name, team).

Thirdly, *semantic relevance* occurs where two objects can be related to each other using some abstraction in the same context i.e. semantic relevance is context dependant e.g. Dept.technicians and Department.students. Fourthly, *semantic resemblance* is the weakest measure of semantic proximity, where the domains of two objects cannot be related to each other in any context e.g. TelecomEmp(ID#, salary) and BankEmp(ID#, salary), where a semantic resemblance could be demonstrated only in limited situations - say where the Inland Revenue were identifying tax liabilities. Finally, *semantic incompatibility* is where there is no context and no abstraction that relates two objects. Therefore, it asserts semantic dissimilarity.

The above approach represents the explicit identification and representation of context between autonomous databases and was proposed as a solution to allow querying of a multidatabase system. Similar work has been demonstrated using Semantic Similarity Relations (SSRs), to define the set of concepts/objects and contexts that provide the basis for a schema comparison stage of an integration process (Song et al., 1996). The relations were classified at four levels defined as: weak semantic relation (where the object pair property sets are overlapping), *compatible semantic relation* (where the property sets of keys overlap), *equivalence semantic relation* (the key property sets are identical) and *mergeable semantic relation* (where property sets are identical).

These relationship classifications were then applied to the various classifications of semantic heterogeneity by using an integrator tool. The tool contained a Prolog-type rule base that served for integration and inferencing, using a semantic dictionary to store words in pairs based on the rules, e.g. Synonym(Client, Customer) or subset[A,C] :- subset[B,C], subset[A,B], and a semantic taxonomy to define hierarchy relationships, such as is_subset of(man, person).

Similar work, involving the Context Interchange (Goh et al., 1999, Firat, 2003), uses a mediator-based approach to analyse the semantics of individual data items, as opposed to conflicts at the schematic level. This differs from integration strategies used in FDBMSs,

which require either administrators or users to resolve conflicts in one or more shared schemas.

These conflict detection approaches form the basis for subsequent development of what have been variously termed common data models (CDM) and summary schema models (Sheth and Larson, 1990, Hammer and McLeod, 1993, Bright, 1994, Bergamaschi et al., 1999, Parent and Spaccapietra, 1998). They facilitate translation of local schema into a canonical or global schema of a FDBMS, using mappings that allow divergent local schemas to be described as a single representation.

7 Discussion

It is recognised that schema design is vulnerable to incompleteness and imprecision because a designer may only emphasise certain aspects of the real world problem. The previous section illustrated the range of structural and semantic conflicts that may be identified during the comparison and conformation stages of integration. Whilst integration approaches have not been the main focus of this review, schema and semantic heterogeneity conflict reconciliation applies to global schema integrations, requiring individual schemas of each database be merged, and FDBMSs that export portions of their schemas for use by the federation users. The issues apply equally to virtualised mediator/wrapper-based approaches.

However, data models address the specific needs and activities of an organisation, and the semantics, conveyed in those models, often represent an *informal* agreement between the developer and department users in a task specific, singular environment. Thus, schemas are not designed to be shareable or reusable, unlike ontology development, which is designed for reuse.

The integration of heterogeneous schemas primarily preserves the structure and integrity of data itself but does not aim to provide *explicit* semantics of the data. Again, ontology structures differ from data models because the fundamental principle of a computing ontology is the representation of information and generic knowledge, through an *agreed* logical view of the domain, achieved by a common vocabulary and definition of concepts and their relationships that can be re-used by different applications. Therefore, there is a greater *formality* in the way in which an ontology represents knowledge, because it is always intended to be a true representation of a domain (Guarino, 1998, Wache et al., 2001).

The Web has provided the platform for an "information space of interrelated resources" (W3C, 2004a) and the Semantic Web (Berners-Lee et al., 2001, Hendler et al., 2002) represents the next generation of the Web "*to create a universal medium for the exchange of data*". Access to billions of diverse information sources, by simple keyword searches, provides limited quality of precision in identifying relevant information. The scale of the integration challenge has changed, requiring the database community to widen its research to encompass all Web content and online databases; to make it easy for everyone to manage most human information online (Bernstein et al., 1998, Gray et al., 2000).

Standardisation at different layers of information systems architectures will be important and it is no coincidence that several enabling key technologies have been adopted as World Wide Web Consortium (W3C) recommendations: the Resource Description Framework (RDF) core language (W3C, 2004c), and the RDF Schema and OWL Web Ontology languages (W3C, 2004b), each co-existing through the syntax of XML.

8 Conclusions

The main objective was to provide an understanding of the causes, types and examples of structural (schema), syntactic (format) and semantic (meaning) heterogeneity, created during the data model and schema design process and, hopefully, this has been achieved.

Integration of heterogeneous databases requires identification and resolution of conflict between schema objects that present similar or equivalent concepts i.e. *semantic reconciliation*. The addition of an ontology layer offers the opportunity to create an overarching contextualisation that can support mappings between related schemas and assist users in domain understanding. An article in Scientific American (Berners-Lee et al., 2001) introduced the Semantic Web as:

"An extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in co-operation".

The exploitation of ontology structures, to provide a solution for online data integration has significant potential. The combination of mediator/wrapper systems, ontologies, and Semantic Web technologies suggest a way forward and will form the main theme of future research.

References

BATINI, C., LENZERINI, M. & NAVATHE, S. B. (1986) A Comparative Analysis of Methodologies for Database Schema Integration. ACM Computing Surveys, 18(4), pp. 323-364.

BERGAMASCHI, S., CASTANO, S. & VINCINI, M. (1999) Semantic Integration of Semistructured and Structured Data Sources. ACM SIGMOD Record - Special Issue on Semantic Interoperability in Global Information, 28(1), pp. 54-59.

BERNERS-LEE, T., HENDLER, J. & LASSILA, O. (2001) The Semantic Web. Scientific American.

BERNSTEIN, P., BRODIE, M., CERI, S., DEWITT, FRANKLIN, M., GARCIA-MOLINA, H., GRAY, J., HELD, J., HELLERSTEIN, J., JAGADISH, H. V., LESK, M., MAIER, D., NAUGHTON, J., PIRAHESH, H., STONEBRAKER, M. & ULLMAN, J. (1998) The Asilomar Report on Database Research. ACM SIGMOD Record, 27(4), pp. 74-80.

BERTINO, E., CATANIA, B. & ZARRI, G. P. (2001) The latest developments - Ch. 4. Intelligent Database Systems. 1st ed. Harlow, UK, Addison-Wesley Professional, pp. 369-403.

BRIGHT, M. W. (1994) Automated Resolution of Semantic Heterogeneity in Multidatabases. ACM Transactions on Database Systems (TODS), 19(2), pp. 212-253.

BRODIE, M. L. (2003) Enterprise Level Integration - It's Not About Technology [online]. Verizon Information Technology. Available from: http://www.plmdc.engin.umich.edu/BrodieIntegration.pdf. [Accessed 29 December 2004].

BUKHRES, O., ELMAGARMID, A., GHERFAL, F. F., LIU, X., BARKER, K. & SCHALLER, T. (1996) The Integration of Database Systems. In BUKHRES, O. & ELMAGARMID, A. (eds.) Object-Oriented Multidatabase Systems. Prentice-Hall, Englewood Cliffs, NJ, pp. 37-56.

CHEN, P. P. S. (1976) The entity-relationship model—toward a unified view of data. ACM Transactions on Database Systems (TODS), 1(1), pp. 9-36.

COLOMB, R. M. (1997) Impact of Semantic Heterogeneity on Federating Databases. The Computer Journal, 40(5), pp. 235-244.

DE BRUIJN, J. (2003) Using Ontologies - Enabling Knowledge Sharing and Reuse on the Semantic Web [online]. Innsbruck, Austria, DERI – Digital Enterprise Research Institute. Available from: http://www.deri.ie/publications/techpapers/documents/DERI-TR-2003-10-29.pdf. [Accessed 15 February 2005].

DREW, P., KING, R., MCLEOD, D., RUSINKIEWICZ, M. & SILBERSCHATZ, A. (1993) Report of the Workshop on Semantic Heterogeneity and Interoperation in Multidatabase Systems. ACM SIGMOD Record, 22(3), pp. 47-56.

ELMASRI, R. & NAVATHE, S. B. (2000) Fundamentals of Database Systems. 3rd ed., Boston, MA: Addison-Wesley Longman, Inc.

FANG, D., HAMMER, J. & MCLEOD, D. (1994) The Identification and Resolution of Semantic Heterogeneity in Multidatabase Systems [online]. Available from: http://citeseer.ist.psu.edu/fang94identification.html. [Accessed 23 November 2004].

FENSEL, D. (2001) Ontologies and Electronic Commerce. IEEE Intelligent Systems, 16(1), pp. 8-14.

FIRAT, A. (2003) Information Integration Using Contextual Knowledge and Ontology Merging. Sloan School of Management - Information Technology Management. Cambridge, MA., Massachusetts Institute of Technology.

GARCIA-MOLINA, H., PAPAKONSTANTINOU, Y., QUASS, D., RAJARAMAN, A., SAGIV, Y., ULLMAN, J. D., VASSALOS, V. & WIDOM, J. (1997) The TSIMMIS approach to mediation: data models and languages. Journal of Intelligent Information Systems, 8(2), pp. 117-132.

GARCIA-SOLACO, M., SALTOR, F. & CASTELLANOS, M. (1996) Semantic heterogeneity in multidatabase systems. In BUKHRES, O. & ELMAGARMID, A. (eds.) Object-Oriented Multidatabase Systems. Prentice-Hall, Englewood Cliffs, NJ, pp. 129-202.

GOH, C. H., BRESSAN, S., MADNICK, S. & SIEGEL, M. (1999) Context Interchange: New Features and Formalisms for the Intelligent Integration of Information. ACM Transactions on Information Systems, 17(3), pp. 270–293.

GRAY, P., ATKINSON, M., GOBLE, C., KAY, M., KERRIDGE, J., MOODY, K. & KING, P. (2000) CPHC Workshop on Research Directions for RAE Themes [online]. Manchester, UK. Available from: http://www.csd.abdn.ac.uk/~pgray/man.html. [Accessed 3 October 2004].

GRUBER, T. R. (1993) A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, 5(2), pp. 199-220.

GUARINO, N. (1998) Formal Ontology and Information Systems. In: Proceedings of 1st International Conference on Formal Ontologies in Information Systems (FOIS'98). Trento, Italy, 6-8 June 1998. IOS Press, pp. 3-15.

HAMMER, J. & MCLEOD, D. (1993) An Approach to Resolving Semantic Heterogeneity in a Federation of Autonomous, Heterogeneous Database Systems. International Journal of Intelligent & Cooperative Information Systems, 2(1), pp. 51-83.

HENDLER, J., BERNERS-LEE, T. & MILLER, E. (2002) Integrating Applications on the Semantic Web. Journal of the Institute of Electrical Engineers of Japan, 122(10), pp. 676-680.

HILL, J. (2004) The Future of Data Integration Technologies [online]. Stamford, CT, META Group Inc. Available from: http://www.dmreview.com/whitepaper/WID1006581.pdf. [Accessed 8 January 2005].

HOWE, D. R. (2003) Data Analysis for Database Design. 3rd ed., Burlington, MA: Butterworth-Heinemann.

KASHYAP, V. & SHETH, A. (1996) Semantic and schematic similarities between database objects: a context-based approach. The VLDB Journal, 5(4), pp. 276-304.

KIM, W. (1991) Object-oriented database systems: strengths and weaknesses. Journal of Object-Oriented Programming, 1(4), pp. 21-29.

KIM, W., CHOI, I., GALA, S. & SCHEEVEL, M. (1993) On Resolving Schematic Heterogeneity in Multidatabase Systems. Distributed and Parallel Databases, 1(3), pp. 251-279.

LAMB, R. & DAVIDSON, E. (2000) The New Computing Archipelago: Intranet Islands of Practice. In: Proceedings of IFIP Working Group 8.2 Conference - The Social and Organizational Perspective on Research and Practice in Information Technology. Aalborg, Denmark, 10-12 June 2000. Kluwer Academic Publishers, pp. 255-274.

LEVY, A., RAJARAMAN, A. & ORDILLE, J. (1996) Querying Heterogeneous Information Sources Using Source Descriptions. In: Proceedings of 22nd International Conference on Very Large Databases (VLDB). Mumbai(Bombay), India, 1996. VLDB Endowment, Saratoga, CA, pp. 251-262.

OUKSEL, A. & SHETH, A. (1999) Semantic Interoperability in Global Information Systems. ACM SIGMOD Record, 28(1), pp. 5-12.

PALOPOLI, L., SACCA, D., TERRACINA, G. & URSINO, D. (2002) A technique for deriving hyponymies and overlappings from database schemes. Data & Knowledge Engineering, 40(3), pp. 285-314.

PARENT, C. & SPACCAPIETRA, S. (1998) Issues and Approaches of Database Integration. Communications of the ACM, 41(5es), pp. 166-178.

PATEL-SCHNEIDER, P. & FENSEL, D. (2002) Layering the Semantic Web: Problems and Directions. In: Proceedings of 1st International Semantic Web Conference (ISWC2002). Sardinia, Italy, 9-12 June 2002.

PITOURA, E., BUKHRES, O. & ELMAGARMID, A. (1995) Object Orientation in Multidatabase Systems. ACM Computing Surveys (CSUR), 27(2), pp. 141-195.

ROB, P. & CORONEL, C. (2002) Database Systems Design, Implementation and Management. Fifth Edition ed.: Course Technology.

SHETH, A. P. (1998) Changing Focus on Interoperability in Information Systems: From System, Syntax, Structure to Semantics. In GOODCHILD, M. F., EGENHOFER, M. J., FEGEAS, R. & KOTTMAN, C. A. (eds.) Interoperating Geographic Information Systems. Dordrecht, Netherlands, Kluwer, Academic Publishers, pp. 5-30.

SHETH, A. P. & KASHYAP, V. (1992) So Far (Schematically) yet So Near (Semantically) [online]. Lorne, Victoria, Australia. Available from: http://lsdis.cs.uga.edu/library/download/SK92b.pdf. [Accessed 11 March 2005].

SHETH, A. P. & LARSON, J. A. (1990) Federated database systems for managing distributed, heterogeneous, and autonomous databases. ACM Computing Surveys (CSUR) - Special issue on heterogeneous databases, 22(3), pp. 183-236.

SONG, W. W., JOHANNESSON, P. & BUBENKO JR., J. A. (1996) Semantic similarity relations and computation in schema integration. Data & Knowledge Engineering, 19(1), pp. 65-97.

STONEBRAKER, M., AGRAWAL, R., DAYAL, U., NEUHOLD, E. J. & REUTER, A. (1993) DBMS Research at a Crossroads: The Vienna Update. In: Proceedings of 19th International Conference on Very Large Databases (VLDB). Dublin, Ireland, 24-27 August 1993. pp. 688-692.

TSENG, F. S. C., CHIANG, J.-J. & YANG, W.-P. (1998) Integration of relations with conflicting schema structures in heterogeneous database systems. Data & Knowledge Engineering, 27(2), pp. 231-248.

W3C (2004a) Architecture of the World Wide Web, Volume One [online]. World Wide Web Consortium. Available from: http://www.w3.org/TR/2004/REC-webarch-20041215/. [Accessed 4 January 2004].

W3C (2004b) OWL Web Ontology Language Guide [online]. World Wide Web Consortium. Available from: http://www.w3.org/TR/owl-guide/. [Accessed 29 October 2004].

W3C (2004c) RDF Primer [online]. World Wide Web Consortium. Available from: http://www.w3.org/TR/rdf-primer/. [Accessed 15 November 2004].

WACHE, H., VÖGELE, T., VISSER, U., STUCKENSCHMIDT, H., SCHUSTER, G., NEUMANN, H. & HÜBNER, S. (2001) Ontology-Based Integration of Information — A Survey of Existing Approaches. In: Proceedings of IJCAI-01 Workshop on Ontologies and Information Sharing. Seattle, WA., 2001. pp. 108-117.

WIEDERHOLD, G. (1992) Mediators in the Architecture of Future Information Systems. IEEE Computer, 25(3), pp. 38-49.