COMPONENT AND FEDERATION CONCEPT MODELS IN A FEDERATED DATABASE SYSTEM

Nayyer Masood
Department of Computer Science
Bahauddin Zakariya University, Multan
PAKISTAN
COMSATS Institute of IT
Wah Cantt, PAKISTAN
email: nayyerm@yahoo.com

Barry Eaglestone
Department of Information Studies
University of Sheffield
Sheffield, S10-2TN, UK
email: beaglestone@sheffeild.ac.uk

ABSTRACT

A methodology for assigning interpretation to a database schema and to develop a federation context in the form of concept models is presented. The concept models are developed using the concepts from an ontology that acts as a semantic fulcrum in the schema integration methodologies. The ontology provides a common vocabulary to establish concept models. The benefit of establishing the concept models is twofold. One, it provides an interpretation to the component schemas of a federated database system. In this perspective it compensates the lack of semantic expressiveness of the current data models. Second, it becomes easier to compare different component schemas and to identify semantic similarities among them that is an essential step in schema integration, hence providing a basis for a semi-automated schema integration approach.

Keywords: ***Schema integration, Schema analysis, Ontology, Concept hierarchies, Schema interpretation, Intrinsic semantics, In-context semantics***

1.0 INTRODUCTION

A multidatabase system aims to provide the access of data from multiple, disparate databases, called the component databases, in a transparent manner. A particular architecture of a multidatabase system is a federated database system (FDBS) proposed in [21]. An FDBS provides the facility of defining different federations of different subsets of database schemas participating/joining the FDBS. The federated schemas are built based on the requirements/interests of different user groups. Fig. 1 shows the 5-level schema architecture of an FDBS.

In 5-level schema architecture, a local schema is basically the conceptual model of a component database expressed in a native data model. So different local schemas may be in different data models. Component schema is the subset of the local schema that the owner organisation is willing to share with other users of the FDBS and it is translated into a common data model. Not all of the component schema may be available to a federation and its users. An export schema represents a subset of a component schema that is available to a particular federation. A federated schema is an integration of multiple export schemas, resulting from the process of schema integration. A federated schema also includes information on the data distribution that is generated when integrating different export schemas. There may be multiple federated schemas in an FDBS, one for each class of federation users. A class of federation users is a group of users and/or applications performing a related set of activities. The component, federated, and external schemas are all in the same data model. External schema is extracted from a federated schema, and is defined for the users/applications of a particular federation [21].

The viability of an FDBS depends on the ability to correctly integrate the component databases into a single (integrated) schema, a process called schema integration (SI). A critical aspect of SI concerns the identification of correspondences among schema elements of CDBSs, that is the elements that are semantically similar. This aspect of SI is problematic because, (i) it is difficult to determine database semantics from database schemas due to lack of semantic expressiveness of the data models. It requires the integrator to have knowledge of concepts relevant to the Universe of Discourse (UoD) of each component databases (CDBS) and the ability to correctly associate these with schema elements that were designed to denote them. In other words, the integrator has to metaphorically "get inside the mind of the designer". (ii) Within the context of FDBSs, similarity between schema elements is often obscured by semantic heterogeneities (SHs), which are a consequence of autonomous design of component databases, and (iii) the task is complex because of the large number of schema element comparisons that must be made [18].

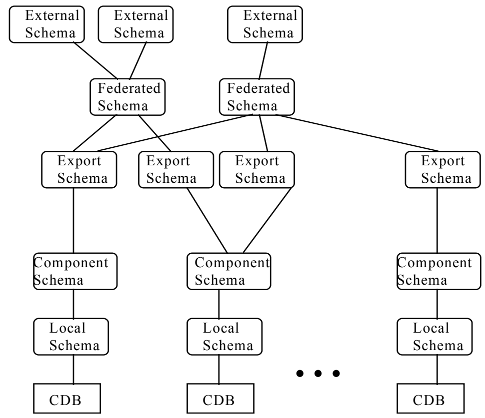

Fig. 1: Five-level Schema Architecture of an FDBS

In this paper we are addressing the first of the three problems in SI, that is, the lack of semantic expressiveness of the component schemas. We are presenting a new methodology used for semantic enrichment of component schemas. Semantic enrichment is necessary because schema integration cannot be based purely on schematic information of component schemas, which currently provide insufficient semantic content to identify and reconcile SHs among them. It is therefore necessary for the analyser to have access to additional semantic information about the component schemas to perform the process.

The semantic enrichment method presented in this paper is an enhancement of the common concept approach [22] and is a part of ECCAM (Extended Common Concept Analysis Methodology) [14]. The major objective of ECCAM is the identification of semantically similar elements among component schemas. These elements can then be integrated to form a federated schema. Hence ECCAM provides the basis of a semantic based schema integration methodology. In ECCAM, first of all we develop an ontology comprising a set of concept hierarchies for the common/global universe of discourse (UoD) being considered in the FDBS. The elements from each component schema are then mapped to the concept(s) (from the ontology) that they model. The mapping process results in a concept model for each component schema called the component concept model (CCM) that serves as its interpretation.

The CCM basically represents the concepts modeled in a component schema and also the perspective of the particular organisation that owns it called the definition context. However, in order to establish a federated schema (Fig. 1) we have to integrate the (subsets of) component schemas in the perspective of a particular federation's users; this perspective is called the federation context. In ECCAM we propose to establish the federation-specific concept model using the concepts from the ontology. This concept model is called federation concept model (FCM). As per the architecture of an FDBS we can have multiple FCMs representing the interests/requirements of different federation's users. An FCM is compared with each CCM in order to identify the elements from a particular component database that model the elements required (to be included) in a particular federated schema. This is major strength of the ECCAM that it uses the concept models to identify semantically similar elements rather than the component schemas, due to which SI process is not affected from semantic heterogeneities.

The structure of this paper is as follows. Section 2 illustrates the mapping of schema elements to concepts that they model; the mapping process assigns concepts to each element, which represent its intrinsic meaning. Section 3 then presents the use of the intrinsic meanings of the schema elements to establish their in-contexts meanings that eventually gives an interpretation of the entire schema. Section 4 discusses the development of federated context for the schema comparison. The related research is discussed in Section 5 and Section 6concludes the paper.

2.0 THE MAPPING PROCESS

This section presents the mapping process in which elements from component schemas are mapped to the concepts in the ontology resulting in the semantic enrichment of the component schemas. This enhancement is basically required for SI, since SI cannot be based solely on the schemas of the component databases due to lack of semantic expressiveness of the current data models. For example, if there is an attribute X in a schema then it would be difficult to know for the integrator what does this attribute mean. In addition to that, SI is also hindered by the existence of semantic heterogeneities (SHs) among component schemas. Following are some examples of the SHs:

- **Naming Conflicts**: The class "PROCESS" meaning a chemical process or a process in a computer memory in two different databases
- **Data Representation Conflicts**: Phone number being represented as a numeric or as a character string in two different databases
- **Attribute Class Conflict**: The "address" being represented as an attribute in one database and as a class in another

There are many other types of SHs that are found among the component databases and they badly hamper the process of SI [15]. The lack of semantic content in the component schemas and the existence of SHs among them necessitate enriching the semantic content of component schemas for SI. In the following we are discussing different phases of our proposed methodology in this regard.

2.1 Developing an Ontology

A pre-requisite to the proposed methodology is the development of an ontology. An ontology is an explicit specification of a conceptualisation where a conceptualisation is an abstract, simplified view of the world that we wish to represent for some purposes [8]. The ontology that is established in the proposed schema interpretation approach consists of general concepts that may exist in the common/global Universe of Discourse (UoD) being modelled within the component databases of an FDBS. The example UoD adopted in this paper is a Library of an educational institution. The concepts in an ontology are arranged in the form of hierarchies, where a concept hierarchy is a tree structure in which the nodes are common/general concepts from the UoD. A concept represents some aspect of reality isolated by mind [1] and is represented by one or more terms. Within the context of the methodology, concept may represent any of the following:

1) a real-world object (physical and/or abstract), like *person*, *book;*
2) any property/characteristic possessed by a real-world object, like *name*, *price*;
3) an activity performed in the UoD, like *issue*, *return* (a book).

Four example concept hierarchies are shown in Fig. 2, where concepts are shown at the nodes and headed arrows represent the is-a relationship among concepts, head pointing towards more general concept. For example, *name is-a reference* is represented in a concept hierarchy by an edge from the *reference* node to the *name* node, with arrow head pointing towards *reference* (more general concept). The link denotes that a *name* is a special type of *reference*. Note however, that the properties of a concept in ontology are abstract since each concept is described only by a simple natural language definition [22].

2.2 Mapping Schema Elements to Concepts

Once the ontology has been established, it can then be used as tool in the mapping process of schema interpretation phase as described below.

The elements from component schemas are mapped to the concepts in the ontology that they model. The aim of the mapping process is to make explicit within the FDBS the meanings or interpretation of the component schemas so that SHs among these schemas can be resolved and semantically similar elements can be identified. The mappings are determined manually by the integrator, as in [22, 19, 16, 7, 2]. This crucial process is complex, since it requires a clear understanding of the semantics of the elements of each component schema, for which the consultation of the local DBAs might be required. This aim is achieved in following three steps:

- Firstly, schema elements are mapped to concepts that represent their **intrinsic semantics**, i.e., (context free semantics). The schema element to concept mappings are defined as a function, *Int*, from schema elements to the power set of concepts in the ontology. Thus, *Int* represents the intrinsic meaning of the schema element O_i, which denotes the concepts, c_i, as

  $$Int\,(O_i) = \{c_{i,}\ i= 1,..,m\}, \qquad \text{(f1)}$$

  where $c_{i,}$, for i = 1, ... m are the concepts denoted by O_i

- Secondly, semantic relationships between schema elements are defined, and are used to establish the **contexts** within which the element is modeled within the schema. The context(s) of a schema element O_i is/are represented by a function, Intr, from the set of schema elements to the power set of 2-tuples, comprising a schema element name and an SR type. Thus, the contexts of a schema element, O_i, are defined as

  $$Intr(O_i) = \{ <O_1, srel_{1i}>, ... < O_n, srel_{ni}>\} \qquad \text{(f2)}$$

where each context, <O_x, $srel_{xi}$>, represents that a semantic relationship of type $srel_{xi}$, represented in the schema, relates O_i to the structural element O_x.

- Thirdly, the intrinsic semantics and the contexts defined for each schema elements in the above two steps are used to establish **in-context semantics** of each element. The in-context semantics of an element denotes its meanings by virtue of structure(s) within which it participates, which are represented as constructs in the schema (Book.title denotes identity of book). The in-context semantics of a schema element O_i in context of element O_x is represented by concatenating the intrinsic meaning of the element O_i, the SR, and the in-context meaning of the element O_{i+1} in context of O_x, i.e.,

  $ICMean(O_i, O_x) = \; <Int(O_i)>$ if $O_i = O_x$
  Otherwise
  $ICMean(O_i,O_x) = ICMean(O_{i+1,x}) \sim srel_{x,x-1} \sim Int(Oi)$

  where $O_x, O_{x-1}, \ldots O_{i+1}, O_i$ denotes the structural path from the in-context schema element O_x to the schema element O_i, and **~** represents the concatenation operation.

For example, consider the following concept hierarchies:

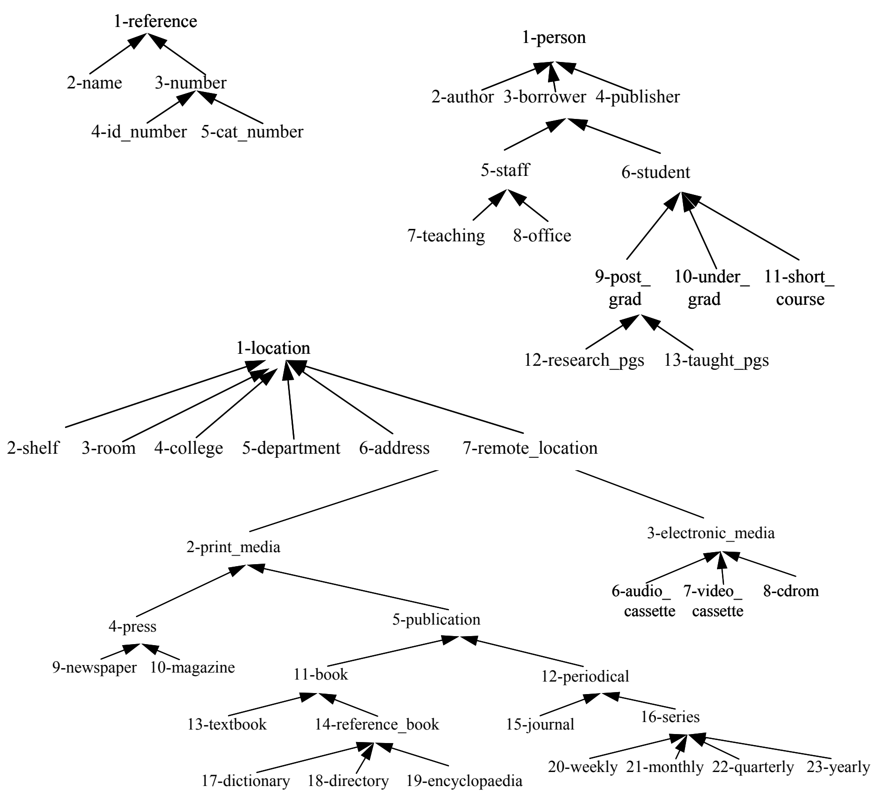

Fig. 2: Example concept hierarchies[1]

Each in-context semantics of an element O_i gives a new/different point of view to O_i; creates a new/different concept. Fig. 3 contains some example class definitions that we have used to give the examples of the ideas presented in this sub-section. The mapping is based on the example concept hierarchies of Fig. 2.

[1] A somewhat complete Ontology is given [M99]

```
class Item
(         extent items)
{           attribute  String  title;
            attribute  Student borrower;
            ......
}
class Person
(         extent persons)
{           attribute  String  name;
            .....
}
class  Student extends Person
(         extent students)
{           attribute  Structure Adres<string hall, string r_num>
                         d_adr;
            .....
}
```

Fig. 3: Some example schema elements

The intrinsic meanings (function *Int)* for some of the schema elements can be defined as:
(a) *Int* (Person) = {teaching_st, office_st, research_pg, taught_pg, under_graduate}
(b) *Int* (Student) = {research_pg, taught_pg, under_graduate}
(c) *Int* (name) = {name}

The contexts of elements (function *Intr*) can be defined as:
(a) *Intr*(title) = = {<Person, has>}
(b) *Intr*(Student)= {<Person, generalises>, <borrower, is-of-type>}

The in-context semantics (*ICMean*(Oi,Ox)) of some elements is given below:
(a) *ICMean* (name, Person) = *Int*(name) ~ has ~ *Int*(Student)
= {*teaching_st, office_st, research_pgs, taught_pgs, under_graduate*}
has {*name*}

(b) *ICMean* (Student, Person) = *Int*(Person) ~ generalises ~ *Int*(Student)
= {*teaching_st, office_st, research_pgs, taught_pgs, under_graduate*}
generalises {*research_pgs, taught_pgs, under_graduate*}

Defining the above mentioned three functions for each schema element of a CDBS provides a basis to establish the interpretation of the entire schema as presented in the next section.

3.0 ESTABLISHING SCHEMA INTERPRETATION

This section discusses the last step of schema interpretation, that is, developing concept model/interpretation of a component schema. The concept model of a component schema consists of the all interpretations of each element in the schema and of SRs between elements. There are three advantages/purposes of developing these concept models in ECCAM:

- different contexts in which an element can be interpreted become evident
- a better understanding of entire schema is obtained
- the concept models of component schemas can be utilised for schema comparison to identify the semantically similar elements among them, hence providing a basis for a semantics based schema integration methodology.

The **concept model** of a schema in our methodology consists of set of two functions, that is, *Int* and *Intr*, defined for each of its constituent element. Both of these functions (f1 and f2) have been defined in the previous subsections.

The function *Intr* is used to establish links of an element with other elements in its immediate contexts. By applying this function iteratively, all the links among the elements of a schema can be established. On the other hand, the set of function *Int* defined for all schema elements represents the intrinsic meanings of the elements. Both of these sets of functions can be utilised:

- to establish in-context semantics of an element within all possible contexts,
- to develop interpretation of entire schema by linking the intrinsic meanings of individual elements through the SRs, and
- in the process of schema comparison to identify the semantically similar elements

In order to fulfill the first two purposes of a CCM, that is, to have a better understanding the semantics of elements and of the entire schema, in a better way, the CCM can also be expressed in the form of a directed graph. In the directed graph, nodes represent schema elements and edges represent the associations, i.e., the SRs, between them. Nodes and edges are labeled. Node labels specify the concepts C_i in the ontology that are modeled by the relevant element O_i, and edge labels describe the type of association represented within the schema.

The Example CCM: In order to demonstrate the schema interpretation phase of the ECCAM, an example is followed in which we have defined a fragment of an example schema from the selected UoD. The example schema is defined in the definition language of the object model 2.0 of the ODMG [4], and is given in Fig. 4.

```
module UniversityLib
class Item
(       extent items) {
        attribute  unsigned short acc_no;
        attribute  String holding;
        attribute  String  title;
}
class  Book extends Item
(       extent books) {
        attribute  set<String>  auth_names;
        attribute  String  c_comp;
        attribute  Publisher  publ;
        relationship Person issued_to inverse Person::bk_issued
}
class  Pap extends Item
(       extent paps) {
        attribute  String  ed_name;
        attribute  String  pap_type;
}
class Person
(       extent persons) {
        attribute  String  name;
        relationship set <Book> bk_issued inverse Book::issued_to;
}
class  student extends Person
(       extent students  {
        attribute  String  reg_no;
        attribute  String  dept;
        attribute  Structure Adress <string  college, string r_num> d_adr;
}
class  Publisher extends Person
(       extent publishers){
        attribute  String  name;
        attribute  String  address;
}
```

Fig. 4: A fragment of an ODMG library object database schema

The intrinsic meanings of some of the elements from above schema using the concept hierarchies of Fig. 2 are shown in Fig. 5.

Int(C_Book) = {textbook}
Int(C_name) = {name}
Int(C_dept) = {name, department}
Int(C_holding) = {id_number, shelf}
Int(C_publ) = {publisher, address}
Int(C_Person) = {author, publisher, teaching_st, office_st, under_graduate, short_course, research_pgs, taught_pgs }

Fig. 5: Int function defined for some of the elements in Fig. 4

The application of the function *Intr*(O_i) on some of the elements of the Fig. 4 is shown in Fig. 6.

Intr(Item) = {}
Intr(Book) = {<*Item*, *generalizes*>}
Intr(Book.title) = {<*Book*, *has*>}
Intr(Book.publ> = { <*Book, has*>}
Intr(Student.d_adr) = {<*Student, lives_at* >}
Intr(Student.d_adr.rnum) = {<*Student.d_adr, has*>}
Intr(Publisher) = {<*Person, generalizes*>,<*Book.publ, is_of-type*>}

Fig. 6: Intr function, capturing the immediate contexts of elements from Fig. 4

All elements of the example component schema linked and interpreted on the basis of their *Intr* functions, the CCM represented as a directed graph is in Fig. 7.

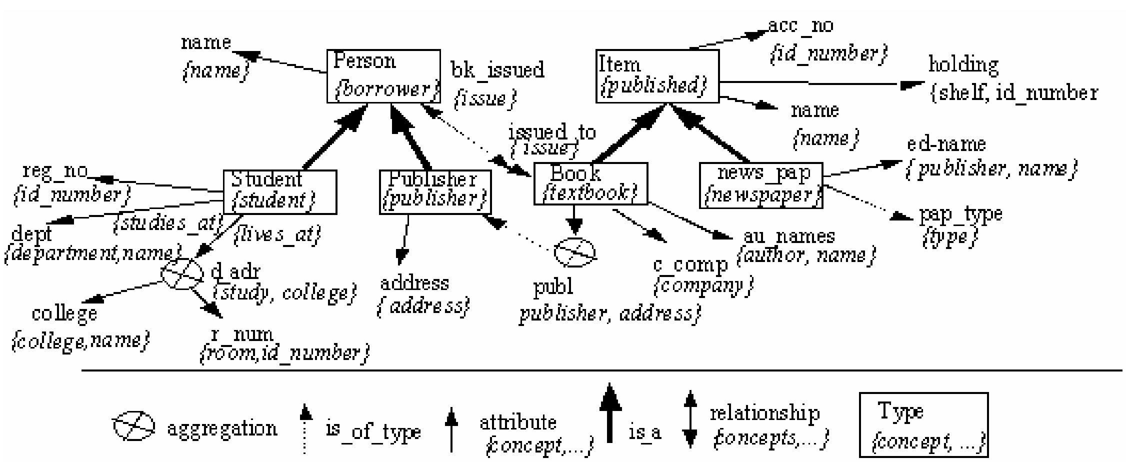

Fig. 7: Directed graph for the schema fragment of Fig. 4

The directed graph for the example component schema that results from its CCM shows:

a) the elements of the schema,
b) the concepts modeled by each schema element (in italics),
c) in-context elements of each schema elements (linked with each other),
d) the SRs between elements; some of the SRs having specific interpretations.

Note that by seeing the CCM in the graphical form as above, one can view all of the different contexts in which each element can be interpreted in a particular schema. This section concludes what is meant by the schema interpretation in the ECCAM. The following section presents the approach adopted to develop the federated concept model in ECCAM.

4.0 FEDERATION CONTEXT FOR COMPONENT SCHEMA COMPARISONS

In the previous section we have shown how an interpretation can be derived for a schema by determining the semantic relationships between schema elements and mapping the elements to the concepts they denote. ECCAM uses the inverse of this approach for determining semantic similarity between schema elements within the context of a specific federation. That is, the interpretation for the federation is defined first and then the federated schema is synthesised such that it conforms to that interpretation.

The semantic similarity between two schema elements depends on the context in which they are being compared, which may be similar to or entirely different from the context in which they are defined [20, 9]. This section describes the proposed approach for the development of a context within which to perform the schema comparison process.

According to ECCAM, the component schemas are not compared with each other to identify the semantically similar elements on the basis of what they model [6, 17, 13, 12, 22], or with a normative global schema as is done in Carnot [5] or with a domain model as in COIN project [2]. Rather, first the context for making such a comparison is established, called the federation context (like the query context in [10]). The base for the federation context is the requirements of particular users' group (federation). The integrator represents the federation context (within which a federated schema is defined) as a federation-specific concept model (FCM). The FCM comprises relevant concepts inter–connected by SRs. Initially, these imply virtual schema elements, since they represent what the specific federated schema is required to represent.

The FCM may be developed either from a schematic or semantic perspective:

- **The semantic approach**: This is a top down approach, in which the integrator first defines the concepts that the target federated schema must denote. Requirements and interests of a federation's users are determined, first. The integrator then selects relevant concepts from the ontology and defines relevant SRs among them. Having established that which must be denoted by the federated schema, the integrator then defines the corresponding (virtual) schema elements, i.e., those implicit in the federated schema's concept model. Thus, the process results in a directed graph that forms the FCM (an example FCM is give in Fig. 8). The concept(s) at each node may be assigned a virtual schema element name that can serve as an actual name in the federated schema when it is established.
- **The schematic approach**: Another way of establishing the FCM, depending upon the requirements of a federation's users, is for the integrator first to define a (virtual) federated schema in which the required schema elements are specified. The process of establishing the concept model for the component schema (Section 3) is then applied to this virtual schema. That is, each virtual schema element is mapped to the concept that it is required to represent. The *Intr* function (see Fig. 6) is then defined for each schema element to determine the SRs in which it is required to participate. As in the previous approach, the result is a directed graph that forms the FCM.

The above process is illustrated below again using the library case study:

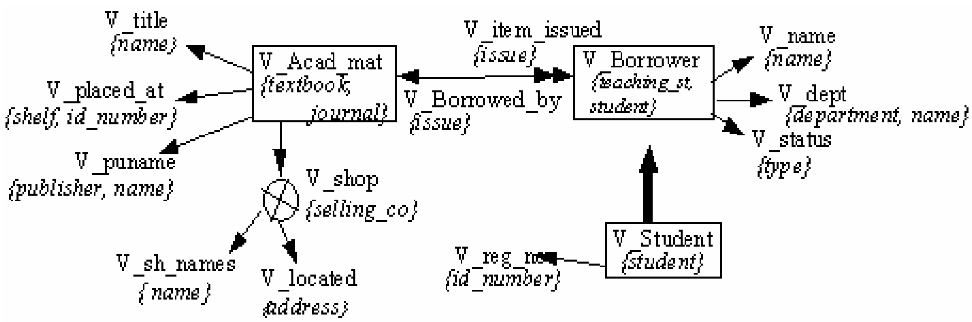

Fig. 8: An example FCM

The figure above shows an example FCM in the form of a directed graph. It could be developed following either of the two approached mentioned above, that is schematic or semantic approach. The 'V_' that precedes every schema element is to show that these are the virtual schema elements; it also helps to distinguish virtual elements from (actual) component schema elements.

Once defined, the FCM provides a basis for identifying the component schema elements that should be integrated to form the federated schema. These are identified by comparing the concepts in the FCM with those denoted by elements of component schemas (from the corresponding CCM). The component schema elements, identified as being relevant are then included within export schemas, so that they can be integrated to form the federated schema. Thus, component schema elements are compared with the virtual elements implied by the FCM, rather than with each other. This reduces the number of comparisons necessary, and eliminates comparisons on the basis of irrelevant concepts.

Construction of the FCM is one of the most crucial steps in ECCAM, since it establishes the basis for identifying component schema elements that must be integrated within the target federated schema. The validity and applicability of the results produced by the schema comparison process directly depends on the validity and completeness of the FCM. The basic purposes of the FCM are therefore:

- to specify the intended meaning of the target federated schema;
- to specify the outline structure of the target federated schema. The FCM specifies a collection of schema elements that must be defined, which are inter-related by SRs that must be represented by appropriate structures, but does not define those structure or the schema element types.
- to specify those concepts that are considered similar within the federation's context.

For example, the virtual schema element V_Acad_mat in Fig. 8 denotes the concepts textbook and journal. This means that later in the schema comparison process any element from the component schemas modelling any or both of the concepts textbook and journal will be considered semantically similar to V_Acad_mat.

5.0 RELATED RESEARCH

This section reviews different approaches of the semantic enrichment of the database schemas. The need for the semantic enrichment of the schemas was realised after consideration of the earlier batch of SI approaches analysed in [3], which have also been termed as the first generation of integration methodologies in [9]. This realisation opens the door to a wave of different schema integration approaches based on adding to the semantic content of the database schemas, and this door is still open.

The theory of attribute equivalence presented in [12] aims to establish different types of relationships among attributes of the component schemas. These relationships later become the basis of establishing correspondence among classes in a bid to integrate them. Equivalence relationships among attributes are based on a set of descriptors defined for each attribute of a component schema. The characteristics represented by descriptors include: uniqueness, lower bound, upper bound, functional dependencies, etc. Each attribute of all the CDBSs is assigned values for these descriptors, which are then compared with each other to determine the equivalence among attributes.

The approach of using of the real world knowledge for the purpose of SI is pursued in the Carnot project [5]. The core of the approach is Cyc [11], a knowledge base that claims to encode the semantics for a significant portion of human consensus reality. It contains the equivalent of 50,000 entities and relationships expressed as frames and slots, and serves as a normative global schema to which all the component schemas are mapped after transforming them to the model used by Cyc. The approach offers a good platform for managing the properties used to represent the semantics of the information resources. The properties include *schema level properties*, like name, domain, format, permissible relationships, documentation etc., and *value level properties* like default value, null, equal (equal property between two objects), inclusion (inclusion property between two objects) etc.

The COntext INterchange (COIN) system [2] adopts a similar approach to ECCAM. In COIN, a domain model is prepared for the domain of interest. The domain model describes the semantics of the "types" of information units and acts as a common vocabulary used in capturing the semantics of the disparate information resources. The domain model, in a sense, describes the domain of interest from the context of the person(s) managing the system (integrator as per the FDBS terminology). The context or semantics of the individual information resources is described with reference to the domain model. This is contrary to ECCAM where component schemas are interpreted using ontology that contains concepts in the domain of interest without any effect of pre-existing point of view or context. So each schema is interpreted purely in the local context. However, for comparing or integrating the component schemas, the federated concept model is prepared that basically represents a particular federation's context.

6.0 CONCLUSION

A major obstacle to the automation of the SI process is the inability to represent real-world semantics within database schemas. This semantic information is necessary to identify semantically similar elements in component schemas. In this paper, we have described a new method for partially overcoming this limitation, whereby the general/common concepts that exist in the global/shared UoD are represented as a set of concept hierarchies, and schemas are given interpretations by mapping elements to concepts within these concept hierarchies. The contribution in this regard has been to extend the mapping approach to take account of higher-level structural elements, the SRs between them, and the development of federation-specific contexts.

The potential advantages of this approach are; (1) the performance of the schema integration process is improved by enabling the matching algorithms to reason within the (real-world) semantic domain, rather than the (computer-world) schematic domain; (2) the persistent concept model and schema element-to-concept mappings ensure that schema interpretations upon which SI is based are consistent between schemas and consistent over time. As a next step, the schema comparison algorithms have to be developed that could use the semantic information made available for each component schema as a result of the work presented in this paper.

REFERENCES

[1] M. Bonjour, G. Falquet, "Concept Base: A Support to Information Systems Integration". *CAISE*94*, Utrecht, 1994.

[2] S. Bressan, C. Goh, N. Levina, S. Madnick, A. Shah, M. Siegel, "Context Knowledge Representation and Reasoning in the Context Interchange System". *Applied Intelligence (13:2)*, September 2000, pp. 165-180.

[3] C. Batini, M. Lenzerini, S. B. Navathe, "A Comparative Analysis of Methodologies for Database Schema Integration". *ACM Computing Surveys*, 18, 4, December 1986, pp. 323-364.

[4] Cattell, R. G. C. ed. *The Object Database Standard: ODMG-93 Release 2.0*. Morgan Kaufman San Francisco, 1996.

[5] C. Collet, M. N, Huhns, W. Shen, "Resource Integration Using a Large Knowledge Base in Carnot". *IEEE Computer*, Dec., 1991.

[6] U. Dayal, H. Hwang, "View Definition and Generalization for Database Integration in a Multidatabase System". *IEEE Transactions on Software Engineering*, November 1984, pp. 629-645.

[7] B. Eaglestone, N. Masood, "Schema Interpretation: An Aid to the Schema Analysis in Federated Database Design". *First International CAiSE'97 Workshop on Engineering Federated Database System*, Barcelona, Spain, June 1997, pp. 1-12.

[8] T. Gruber, "A Translation Approach to Portable Ontology Specification". *Knowledge Acquisition, An International Journal of Knowledge Acquisition for Knowledge-Based Systems*, 5, 2, June 1993.

[9] M. Garcia-Solace, F. Saltor, M. Castellanos, "Semantic Heterogeneity in Multidatabase Systems". *Object-Oriented Multidatabase Systems A Solution for Advanced Applications*, Bukhres, O. A., Elmagarmid, A., Chapter 5, PHI, 1996.

[10] V. Kashyap, A. Sheth, "Semantic and Schematic Similarities between Database Objects: A Context Based Approach", *22nd VLDB*, Bombay, 1996.

[11] D. Lenat, R. V. Guha, *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project.* Addison-Wesely, Reading, Mass., 1990.

[12] J. A. Larson, S. B. Navathe, R. Elmasri, "A Theory of Attribute Equivalence in Database with Application to Schema Integration". *IEEE Transactions on Software Engineering*, April 1989, pp. 449-463.

[13] A. Motro, “Superviews: Virtual Integration of Multiple Databases”. *IEEE Transactions on Software Engineering*, 13(7), 1987, pp. 785-798.

[14] N. Masood, “Semantics Based Schema Analysis”. *Ph.D. Dissertation*, University of Bradford, UK, February 1999.

[15] N. Masood, “Semantic Heterogeneities: An Impedance to the Interoperability among Databases”. *Accepted in Karachi University Journal of Science*, University of Karachi, Pakistan, April 2000.

[16] E. Mena, K. Kashyap, A. Sheth, A. Illaramendi, “OBSERVER: An Approach for Query Processing in Global Information Systems Based on Interoperation Across Pre-Existing Ontologies”, in *Proceedings of the First IFCIS Conference on Cooperative Information Systems (COOpIS’96)*, Bressels, Belgium, June 1996.

[17] S. Navathe, R. Elmasri, J. Larson, “Integrating User Views in Database Design”. *IEEE Computer*, 19(1), January 1986, pp. 50-62.

[18] S. Navathe, A. Savasere, “A Schema Integration Facility Using Object-Oriented Data Model”, in *Object-Oriented Multidatabase Systems: A Solution for Advanced Applications*, PHI, 1996.

[19] A. P. Sheth, S. K. Gala, S. B. Navathe, “On Automatic Reasoning for Schema Integration”. *International Journal of Intelligent Co-operative Information Systems*, 2(1), March 1993, pp. 23-50.

[20] A. Sheth, V. Kashyap, “So Far (Schematically) Yet So Near (Semantically)”, in *Proceedings, IFIP Wg 2.6 Conference on Semantics of Interoperable Database Systems (Data Semantics 5)*, North Holland, Amsterdam, 1993, pp. 283-312.

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

[22] C. Yu, W. Sun, S. Dao, Keirsey, D., “Determining Relationships Among Attributes for Interoperability of Multi-Databases Systems”, in *Proceedings of the First International Workshop on Interoperability in Multidatabase Systems,* IMS, Kyoto, Japan, April 1991, pp. 251-257.

BIOGRAPHY

Nayyer Masood did his Master degree from Quaid-e-Azam University Islamabad, Pakistan in 1988. Then he joined Bahauddin Zakariya University Multan as lecturer. He then did his Ph.D. from Bradford University, UK in 1999. Currently he is working in COMSATS Institute of IT in Wah Cantt as Associate Professor. His research areas include schema comparison, semantic web and grid computing.

Barry Eaglestone is currently working as Senior Lecturer in department of Information studies at University of Sheffield, UK. He did his B.Sc. from Stafford and PhD from Huddersfield. His teaching interests include relational and object database systems and design methods; information systems modeling; information management and technology in healthcare. His research interests are in the areas of data models for advanced applications, particularly those characterized by temporal information; Web-database transactions; database support for computer music; federated database systems, particularly schema integration and schema evolution; use of neural networks within database systems; healthcare applications of database systems. Barry Eaglestone has published his work in many conferences and journals and has also authored many books.