▼ Journal of Research (Science), Bahauddin Zakariya University, Multan, Pakistan.
Vol.14, No.2, December 2003, pp. 213-223 ISSN 1021-1012

SHALLOW AND DEEP SEMANTIC SIMILARITY AMONG SCHEMA ELEMENTS

Nayyer Masood
Department of Computer Science, Bahauddin Zakariya University, Multan, Pakistan.
email: nayyerm@yahoo.com

Abstract: Semantic similarity between schema elements is greatly influenced by the context in which the elements are defined and compared. This paper emphasizes on the role of context in establishing semantic similarity between schema elements resulting two different forms of semantic similarity, i.e., shallow similarity and deep similarity. Shallow similarity is based on the inherent meanings of the elements only, where as deep similarity is a context based semantic similarity. The proper description of semantic similarity is helpful in identifying the corresponding schema elements for the purpose of schema integration. A new taxonomy of semantic similarity presented in this paper also helps to identify the exact nature of correspondence among schema elements, which helps the integrator to determine exact treatment for the corresponding schema elements in schema integration.

Keywords: Context, schema integration, semantic similarity, semantics.

INTRODUCTION

Integrating data from multiple disparate databases requires establishing correspondences among the schema elements from these databases. The schematic contents of the database schemas are not enough to establish such correspondences among schema elements rather they have to be established on the basis of databases' semantics [Garcia-Solace *et al.* 1996]. The corresponding schema elements are said to be semantically similar, that is, modeling the same or similar concepts. The semantically similar schema elements are then candidates for being merged into an integrated schema. The exact form of schema elements integration is based on the nature of semantic similarity among them. In other words, knowing the form of semantic similarity is critical to integrate the schema elements properly.

ECCAM (Extended Common Concept based Analysis Methodology) is a semantic-based schema analysis methodology that aims to identify the semantic similarity among schema elements of different databases [Masood 1999]. Semantic similarity between two schema elements is influenced by two main factors; the concept these schema elements are modeling and the context in which they are being compared [Garcia-Solace *et al.* 1996]. Knowing the concept(s) being modeled by a schema element clarifies the inherent meaning of the element, whereas the context describes the point of view or interest of the organization in modeling that element. ECCAM identifies semantically similar elements using both of the above factors; it results different forms of semantic similarities giving rise to a taxonomy that I am presenting in this paper.

The structure of the paper is as follows: Section 2 (Schema Semantics) presents what I mean by semantics in this paper. Next in section 3 (Semantic Similarity) I have discussed two basic forms of semantic similarity, that is, the shallow similarity and deep similarity. Section 4 (Taxonomy of Semantic Similarity) presents taxonomy of deep similarity that provides a basis for identifying the exact nature of correspondence between schema elements. The related work and concluded remarks are given at the end.

SCHEMA SEMANTICS

In this section I am briefly re-stating the ECCAM's approach towards schema semantics that has been discussed in detail in [Masood 1999]. Once different forms of semantics are in mind it would become easier to understand the proposed taxonomy of semantic similarity.

INTRINSIC AND IN-CONTEXT SEMANTICS OF SCHEMA ELEMENTS

The intrinsic semantics of a schema element is its meaning, i.e., the concepts that it denotes, independent of the context within which it is used. Formally, intrinsic semantics is defined as:

> The **intrinsic semantics** of a schema element is represented by the function *Int* from the set of schema element names to the power set of concepts. Thus, the intrinsic semantic of a schema element O_i, is defined by
>
> $$Int\ (O_i) = \{c_{i,}\ i= 1,m\}, \qquad \text{(f-1)}$$
>
> where c_i, for i = 1, ... , m are the concepts denoted by O_i

Further, the **in-context semantics** of a schema element are its more specific semantics within the contexts in which the element is defined. The in-context semantics of a schema element are determined by the concepts that it denotes, i.e., its intrinsic semantics, and the **contexts** within which it is modeled.

The contexts of a schema element are modeled within a schema by the structures in which the element is defined, which in turn denote semantic relationships (SRs) between the schema element and the structural elements to which it is related. Formally, a context of a schema element is defined as follows:

> The context of a schema element O_i with respect to a structural schema element O_x is defined by the function, *context*, as:
>
> $context\ (O_i,O_x) = <O_x>\ where\ Oi = Ox$, otherwise
>
> $context\ (O_i,O_x) = O_x\ srel_{x,\ x-1}\ context(O_i,\ Ox\text{-}1)$
>
> where $O_x,O_{x-1},\ldots O_{i+1},O_i$ denoted the elements in the structural path from the context schema element Ox to the schema element Oi, linked with each other through SRs $srel_{x,x-1}$, …., $srel_{i+1,i}$.

A *context*(O_i, O_x) is an **immediate context** of O_i if O_i and O_x are linked with each other through a single SR, for example, Person has name, Book is-a Item etc.

A schema element may be interpreted as having different specific semantics, depending upon the contexts within which it is defined. I call these context-specific interpretations of a schema element, its **in-context semantics**.

The **in-context semantics** a schema element O_i in *context*(O_i, O_x) is represented by concatenating the intrinsic meaning of the element Oi, the SR $srel_{x,x-1}$, and the in-context meaning of the element O_{i+1} in *context* (O_{i+1}, O_x), i.e.,

$$ICMean(O_i, O_x) = \langle Int(O_i) \rangle \text{ if } O_i = O_x, \quad \text{Otherwise}$$

$$ICMean(O_i, O_x) = ICMean(O_{i+1,x}) \frown srel_{x,x-1} \frown Int(O_i)$$

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.

Following figure contains some example schema elements along with their intrinsic semantics followed by their in-context semantics:

Elements' Definitions	Intrinsic meanings of elements
class Item { ..... }	*Int*(Item) = {textbook, reference_book, journal, series}
class Book **extends** Item { **attribute** String title; **attribute** set<String> auth_names; }	*Int*(Book) = {textbook} *Int*(title) = {name} *Int*(auth_names) = {author, name}
class Acad_mat { **attribute** String name; **attribute** Publisher publ; }	*Int*(Acad_mat) = {textbook} *Int*(name) = {name} *Int*(publ) = {publisher}

Fig. 1: Schema elements and their respective intrinsic semantics.

In-context semantics of some of the elements from Fig. 1 are:

$$ICMean(\text{title, Book}) = Int(\text{Book}) \frown \text{has} \frown Int(\text{title})$$
$$= \{textbook\} \text{ has } \{name\}$$

$$ICMean(\text{Book, Item}) = Int(\text{Item}) \frown \text{generalizes} \frown Int(\text{Book})$$
$$= \{\text{textbook, reference\_book, journal, series}\} \text{ generalizes } \{textbook\}$$

SEMANTIC SIMILARITY

This section presents the approach adopted towards establishing semantic similarity in ECCAM. Firstly, two main forms of semantic similarity are identified, that is, the shallow and deep similarities. Shallow similarity is then used in establishing different forms of the deep similarity.

SHALLOW SIMILARITY

Shallow similarity between a pair of schema elements is based on their respective intrinsic meanings only (section 2). Two schema elements (O_i, O_j) have **shallow similarity** between them if there are some concepts common among their intrinsic meanings. The existence of shallow

similarity between two elements can be determined using the *Int* function (f-1). Formally, I define shallow similarity as follows:

Two schema elements, O_i and O_j, are **shallow similar** if:

$$Int(Oi) \cap Int(Oj) \neq \varnothing$$

For example, consider the elements' definitions and their respective *Int* function definitions in Fig. 1. On the basis of *Int* functions defined above the following relationships hold,

a) $(Int(\text{Acad\_mat}) \cap Int(\text{Item})) \neq \varnothing$
b) $(Int(\text{title}) \cap Int(\text{Item})) = \varnothing$
c) $(Int(\text{Acad\_mat}) \cap Int(\text{Book})) \neq \varnothing$

Therefore, the following declarations about shallow similarities can be made:

1. Elements in statements (a) and (c) are shallow similar since the intersection of their intrinsic meanings are not null.
2. The elements pairs in statement (b) are not shallow similar due to a null intersection between intrinsic meanings.

Shallow similarity between schema elements does not by itself provide a sufficient basis for merging the schema elements, since elements with the same intrinsic meanings can be semantically distinct within the contexts in which they are defined. However shallow similarity of schema elements provides a basis for identifying the similarities that may exist between them within those contexts, i.e., the in-context semantics. The latter is then the basis for asserting semantic similarity between schema elements and merging them, within an integrated schema.

DEEP SIMILARITY

The second type of semantic similarity defined in this paper is deep similarity. It is normally known as semantic similarity in the literature, but has been given this specific name to distinguish it from shallow similarity, defined in the previous section. From this point on the term semantic similarity also means the deep similarity unless stated otherwise.

Schema elements cannot be integrated on the basis of shallow similarity alone, because the same concept(s) may represent different real-world objects when viewed in different contexts. For instance the intrinsic meanings of attributes, Item.title and Person.name may represent the same concept *name.* On this basis the two attributes are semantically similar, but when viewed in the context of the respective classes in which they participate, semantic differences become apparent. Eligibility of two schema elements to be integrated cannot therefore be asserted on the basis of shallow similarity between them alone.

The definitions of intrinsic and in-context semantics (section 2), and of shallow similarity form the basis of the definition and classification of semantic similarity used in ECCAM. Informally, I define semantic similarity between schema elements as follows:

Two schema elements are semantically similar if there is a correspondence between the concepts that they model when compared in a particular context.

Specifically, semantic similarity is defined for contexts that have compatible structures. By this, I mean:

- They comprise the same number of structural elements,
- Corresponding SRs are compatible, i.e., they can be merged into a single SR. Compatibility between two SRs is a problem only if either or both of them have been assigned an interpretation during schema interpretation phase of ECCAM [Masood 1999], in which case their respective interpretations have to be analyzed to determine the compatibility between them. For example the attribute Staff.adr; the SR between attribute adr and the class Staff is "has". This SR may be interpreted as "residential address" or "office address" or "previous address" etc. Yet the attribute of a similar class in another database can be interpreted differently, so WE have to consider the nature of SRs between two schema elements also to reach any decision regarding their integration. If neither of the two SRs has an interpretation then they can always be merged into a single SR, even if they are structurally different.

Semantic similarity is therefore formally defined as follows:

Semantic similarity between a pair of elements O_i and O_j, with respect to the contexts, *context*(O_i, O_x) and context(O_j, O_y), exists if the following conditions hold:

1- $Int(O_i) \cap Int(O_j) \neq \varnothing$, i.e., O_i and O_j are shallow similar

2- if ($O_i \neq O_x$) and ($O_j \neq O_y$) then
$srel_{i+1,i} \approx srel_{j+1,j}$ i.e., the corresponding SRs are compatible

3- O_{i+1} and O_{j+1} are semantically similar respectively within *context*(O_{i+1},O_x) and *context*(O_{j+1},O_y)

where O_x,O_{x-1},…O_{i+1},O_i denotes the structural path from the context element O_x to the schema element O_i, and O_y,O_{y-1},…O_{j+1},O_j denotes the structural path from the context element O_y to the schema element O_j.

For example, consider the elements pair (name, title), respectively in the contexts (name, Acad_mat) and (title, Book) given in Fig. 1 above. The following can be noted about these elements:

1- (*Int*(name) $\cap$ *Int*(title)) $\neq \varnothing$

2- Has $\approx$ Has

3- (*Int*(Acad_Mat) $\cap$ *Int*(Book)) $\neq \varnothing$

Since all three conditions for semantic similarity are satisfied, elements name and title are semantically similar to each other in contexts (name, Acad_mat) and (title, Book).

THE TAXONOMY OF SEMANTIC SIMILARITY

This section presents taxonomy of semantic similarities that may exist between schema elements. The taxonomy presented in this paper classifies semantically similarity between schema element pairs as being semantically equivalent, related or disjoint as given in [Yu *et al*. 1999] and also contextually disjoint. These three categories respectively reflect strong, weak and an absence of semantic similarity between elements, or a lack of any semantic similarity between the contexts. These classes are defined as follows:

SEMANTICALLY EQUIVALENT SCHEMA ELEMENTS

Two schema elements, O_i and O_j, are said to be semantically equivalent if they model exactly the same concepts in a particular context. Formally, I define this as follows:

Two schema elements O_i and O_j, in the contexts (O_i, O_x) and (O_j, O_y) are **semantically equivalent** if the following conditions hold:

1- $Int(O_i) = Int(O_j)$, i.e., O_i and O_j model exactly the same concepts

2- if ($O_i \neq O_x$) and ($O_j \neq O_y$) then $srel_{i+1,i} \approx srel_{j+1,j}$

3- O_{i+1} and O_{j+1} are semantically equivalent respectively within $context(O_{i+1},O_x)$ and $context(O_{j+1},O_y)$

where $O_x,O_{x-1},\ldots O_{i+1},O_i$ denotes the structural path from the context element O_x to the schema element O_i, and $O_y,O_{y-1},\ldots O_{j+1},O_j$ denotes the structural path from the context element O_y to the schema element O_j.

That is, two schema elements (O_i, O_j) are semantically equivalent contexts (O_i, O_x) and (O_j, O_y), if

(a) All the corresponding schema elements involved in the path names between O_i, O_x and O_j, O_y model exactly the same concepts, that is, their intrinsic meanings are exactly the same, and

(b) The corresponding SRs linking O_i, O_x and O_j, O_y are compatible with each other.

For example, if we consider the schema elements, name and title, from the figure 1, the following relationships hold:

1- (*Int*(name) = *Int*(title)), i.e., {name} = {name}

2- Has ≈ Has

3- (*Int*(Acad_Mat) = *Int*(Book)), i.e., {textbook} = {textbook}

Therefore, the elements name and title with context elements Acad_mat and Book are actually semantically equivalent.

SEMANTICALLY RELATED SCHEMA ELEMENTS

Two schema elements (O_i, O_j) are semantically related if they have some concept(s) in common a particular context.

Two schema elements O_i and O_j, in the contexts (O_i, O_x) and (O_j, O_y) are **semantically related** if they are not semantically equivalent (defined above) and the following conditions hold:

1- $Int(O_i) \cap Int(O_j) \neq \varnothing$, i.e., some (not all) concepts among O_i and O_j are common

2- if ($O_i \neq O_x$) and ($O_j \neq O_y$) then
$srel_{i+1,i} \approx srel_{j+1,j}$

3- O_{I+1} and O_{j+1} are semantically related respectively within $context(O_{i+1},O_x)$ and $context(O_{j+1},O_y)$

where $O_x,O_{x-1},\ldots O_{i+1},O_i$ denotes the structural path from the context element O_x to the schema element O_i, and $O_y,O_{y-1},\ldots O_{j+1},O_j$ denotes the structural path from the context element O_y to the schema element O_j.

The above definition means that if

(a) All the elements involved in path name between O_i, O_x and O_j, O_y have some concepts common between their intrinsic meanings, and all of them are not exactly the same, and

(b) The SRs involved are compatible

then the elements are semantically related to each other.

The elements (name, auth_name) with respective context elements Acad_mat and Book in Fig. 1 are not semantically equivalent (Int(name) $\neq$ Int(auth_name), but

1- Int(name) $\cap$ Int(auth_name) $\neq \varnothing$ and Int(name) $\neq$ Int(auth_name)

2- Has $\approx$ Has

3- (Int(Acad_Mat) = Int(Book)), i.e., {textbook} = {textbook}

therefore elements (name, auth_name) are semantically related.

SEMANTICALLY DISJOINT SCHEMA ELEMENTS

According to ECCAM two schema elements (O_i, O_j) are semantically disjoint if there is no concept common between their intrinsic meanings, that is:

Two schema elements O_i and O_j are **semantically disjoint** if the following conditions hold:

$Int(Oi) \cap Int(Oj) = \varnothing$

For example, amongst the schema elements shown in Fig. 1 above the elements publ and auth_names are disjoint, since

Int(publ) $\cap$ Int(auth_names) $= \varnothing$, i.e., {publisher} $\cap$ {author, name} $= \varnothing$

The semantic disjoint schema elements are defined on the basis of intrinsic meanings only, because if two elements (O_i, O_j) have some concept common between their respective intrinsic meanings, i.e., have shallow similarity then this indicates the existence of some sort of semantic relationship among them. However, if they do not have shallow similarity then they are semantically disjoint.

CONTEXTUALLY DISJOINT SCHEMA ELEMENTS

This category includes those element pairs, which are neither semantically equivalent, related nor disjoint. These are the pairs, which are intrinsically shallow, similar, but cannot be related because there is no correspondence between their respective contexts.

Two schema elements O_i and O_j are **contextually disjoint** if the following conditions hold:

1- $Int(O_i) \cap Int(O_j) \neq \varnothing$, i.e., some or all concepts among O_i and O_j are common

2- if $(O_i \neq O_x)$ and $(O_j \neq O_y)$ then
NOT$(srel_{i+1,i} \approx srel_{j+1,j})$ OR $(Int(O_x) \cap Int(O_y) = \varnothing)$

where O_x and O_y represent any of the immediate context elements of O_i and O_j respectively.

The above conditions represent a situation when two elements O_i and O_j have nonzero shallow similarity, but none of their immediate context elements have nonzero shallow similarity, or none of the SRs involved in the immediate contexts of O_i and O_j are compatible.

RELATED WORK

Taxonomy of semantic similarity is critical to a schema analysis approach since it is used to classify the similarity established among elements during schema analysis. That is why we see different taxonomies defined in different SI approaches [Dayal and Hwang 1984, Navathe *et al*. 1986, Larson *et al.* 1989, Yu *et al.* 1991, Sheth and Kashyap 1993, Sheth *et al.* 1993, Eaglestone and Masood 1997]. On the other hand, semantics of the elements and semantic similarity among elements is treated differently by different SI/schema analysis approaches, so a general taxonomy cannot strictly be adopted by different schema analysis approaches. Since ECCAM is based on new ideas regarding semantic similarity, it has also to adopt its own taxonomy of semantic similarity; the types which the ECCAM will classify the similar elements into.

The taxonomy of the semantic similarity defined in the previous section has similarities with other taxonomies in the literature. For example, both the proposed taxonomy and the one defined in [Yu et al. 1991] define equivalent, related and disjoint semantic similarities. However, there are number of differences which make the proposed taxonomy a stronger basis for schema analysis. For example, the approach adopted for mapping schema elements to the corresponding concepts is different in the two; and the similarity is computed in a certain context in the ECCAM, whereas context is not considered in the similarity computation process in previous approaches.

There are also considerable differences in the taxonomy presented in this paper and those in [Larson *et al.* 1989, Kashyap and Sheth 1996]. These differences concern the conceptual level at which the semantic similarities are defined, and the specificity of the definitions.

CONCEPTUAL LEVEL

In general, a major characteristic that differentiates the definitions of semantic similarity proposed in this paper from those in other taxonomies is that, in our taxonomy the similarity is considered purely at the conceptual level. That is, the definition is based upon the mappings from schema elements to concepts [Masood 1999, Masood 2000], which makes explicit the semantics of each schema element.

In contrast, the definitions of Strong, Weak, and Disjoint semantic similarity in Larson *et al.* [1989] are based on the existence of mapping between the domains of the attributes, rather than their meanings. Consequently, semantic similarity is mainly based on aspects that are a consequence of schema element semantics, rather that directly on the semantics themselves.

In the taxonomy presented by Kashyap and Sheth [1996], similarities among schema elements are defined on the basis of both the semantic aspects of the elements and also those aspects, which are an indirect consequence of these. The latter is involved as the similarity computation involves the comparison of schema elements' domains as given by Larson *et al.* [1989]. However, semantic aspects are also involved in this taxonomy as both the definition and the query contexts are considered in the definition of semantic similarity between schema elements. Contexts are defined by using dynamically defined sets of descriptors/meta-attributes from an ontology. The similarity computation involves the comparison of values of descriptors/meta-attributes to identify the correspondence between the semantics contents (integrity constraints etc.) of the elements. There are three reasons due to which I have not adopted this particular taxonomy in ECCAM:

- ECCAM establishes semantic similarity among elements on the basis of their intrinsic and in-context semantics separately. This, in particular, leads to the definition of contextually disjoint schema elements. The separation of intrinsic and in-context semantics of schema elements is not possible in the approach of Kashyap and Sheth [1996]
- Other than *semantic equivalence* and *semantic incompatibility*, [Kashyap and Sheth 1996] define three types of semantic similarity, which include *semantic resemblance*, *semantic relevance*, and *semantic relationship*. The former two types are same as *semantic equivalence* and *semantic disjoint* defined in this thesis, but the latter three are not useful/required from ECCAM point of view. According to ECCAM, elements that fall in latter three categories of Kashyap and Sheth [1996] are declared as *semantically related* with a numeric value between 0 and 1 showing the level of similarity between them
- Finally, there is a basic difference in the approach of establishing semantic similarity, which is based on both the structural and

semantic aspects of elements in the previous approach, where as ECCAM does it only on the semantic aspect of schema elements.

CONCLUSION

In this paper, I have presented a new perspective to define and use semantic similarity. This perspective is based on a very critical aspect that the semantic similarity between elements is based on the context in which elements are compared. This aspect is used to present different types of semantic similarity, that is, the shallow and the deep similarity. A new taxonomy of the semantic similarity is also presented that establishes the basis for integration of schema elements. The next phase in ECCAM is to devise a comparison approach that makes use of the concept models presented by Masood [1999] to establish the semantic similarities presented in this paper. The elements identified as semantically similar can then be merged into an integrated schema.

References

Dayal, U. and Hwang, H. (**1984**) "View definition and generalization for database integration in a multi-database system", *IEEE Transactions on Software Engineering*, November 1984, 629-645.

Eaglestone, B. and Masood, N. (**1997**) "Schema interpretation: An aid to the schema analysis in federated database design", *Proceedings of First International CAISE'97 Workshop on Engineering Federated Database System*, Barcelona, Spain, 1-12.

Garcia-Solace, M., Saltor, F. and Castellanos, M. (**1996**) "Semantic Heterogeneity in Multidatabase Systems", In: O.A. Bukhres and A. Elmagarmid (Eds.), *Object-Oriented Multi-database Systems: A Solution for Advanced Applications*, PHI, New Jersey, pp. 129-193.

Kashyap, V. and Sheth, A. (**1996**) "Semantic and schematic similarities between database objects: A context based approach", *Proceedings of 22nd VLDB*, Bombay, India.

Larson, L.A., Navathe, S.B. and Elmasri, R. (**1989**) "A theory of attribute equivalence in database with application to schema integration", *IEEE Transactions on Software Engineering*, April 1989, 449-463.

Masood, N. (**1999**) "Semantics Based Schema Analysis", Ph.D. Thesis, Department of Computing, University of Bradford, UK

Masood, N. (**2000**) "Semantic heterogeneities: An impedance to the interoperability among databases", *Karachi University Journal of Science*, University of Karachi, Pakistan, 29(1&2), 79-92.

Navathe, S.B., Elmasri, R. and Larson, J. (**1986**) "Integrating user views in database design", *IEEE Computer*, 19(1), 50-62.

Sheth, A. and Kashyap, V. (**1993**) "So far (schematically) yet so near (semantically)", *Proceedings of IFIP Wg 2.6 Conference on Semantics of Interoperable Database Systems (Data Semantics 5)*, North Holland, Amsterdam, 283-312.

Sheth, A., Gala, S.K. and Navathe, S.B. (**1993**) “On automatic reasoning for schema integration”, *International Journal of Intelligent Cooperative Information Systems*, 2(1), 23-50.

Yu, C., Sun, W., Dao, S. and Keirsey, D. **(1991)** “Determining relationships among attributes for interoperability of multi-databases systems”, *Proceedings of the first International Workshop on Interoperability in Multi-database Systems,* Kyoto, Japan, 251-257.