From Proc. ER05
LNCS

Schema Integration based on Uncertain Semantic Mappings

Matteo Magnani[1], Nikos Rizopoulos[2], Peter McBrien[2], and Danilo Montesi[3]

[1] Department of Computer Science, University of Bologna
Via Mura A.Zamboni 7, 40127 Bologna, Italy
matteo.magnani@cs.unibo.it
[2] Department of Computing, Imperial College London
180 Queen's Gate, South Kensington Campus, London SW7 2AZ, UK
{nr600,pjm}@doc.ic.ac.uk
[3] Department of Mathematics and Informatics, University of Camerino
Via Madonna delle Carceri 9, I-62032 Camerino (MC), Italy
danilo.montesi@unicam.it

Abstract. Schema integration is the activity of providing a unified representation of multiple data sources. The core problems in schema integration are: *schema matching*, *i.e.* the identification of correspondences, or *mappings*, between schema objects, and *schema merging*, *i.e.* the creation of a unified schema based on the identified mappings. Existing schema matching approaches attempt to identify a single mapping between each pair of objects, for which they are 100% certain of its correctness. However, this is impossible in general, thus a human expert always has to validate or modify it. In this paper, we propose a new schema integration approach where the uncertainty in the identified mappings that is inherent in the schema matching process is explicitly represented, and that uncertainty propagates to the schema merging process, and finally it is depicted in the resulting integrated schema.

1 Introduction

In this paper we present a new method of schema integration based on uncertain semantic mappings. Schema integration is the activity of providing a unified representation of multiple data sources. The core problems in schema integration are: *schema matching* [1], *i.e.* the identification of correspondences, or *mappings*, between schema objects, and *schema merging* [2], *i.e.* the creation of a unified schema based on the identified mappings. In our approach, we focus on semantic schema integration and on semantic mappings between schema objects. Knowledge about semantic mappings is essential to produce an integrated schema [3]. Early [6, 7] and more recent work [4, 5, 8] has shown that if all semantic mappings are known, then schema merging can be performed semi-automatically.

Unfortunately, it can be very difficult to identify semantic mappings with certainty. Manual schema matching is usually time consuming, and it may be unfeasible, especially with large databases. Automatic schema matching is inherently uncertain because the semantics of schema objects cannot be fully derived

from data and meta-data information. In our novel approach, uncertainty in the identified mappings is represented during the schema matching process, that uncertainty propagates to the schema merging process, and it is depicted in the resulting integrated schema.

As a motivating example, consider the schemas S_1 and S_2 in Figure 1. Schema S_1 models a data source of undergraduate students. Undergraduates are registered (reg) in courses that are taught (tch) by staff members. Schema S_2 models a data source of postgraduate students, which can also optionally register in fourth-year undergraduate courses to refresh their knowledge or familiarize themselves with new subjects. Therefore, S_1.student and S_2.student are disjoint, while S_1.course subsumes S_2.course. Additionally, S_1.staff and S_2.staff are equivalent. The cardinalities of the tch relationship in the two schemas differ, since not all staff members teach fourth-year courses. The aforementioned semantic mappings drive the schema merging process. For example, the disjointness mapping between the student entities triggers schema transformations that rename the entities to make them distinct, *e.g.* into ug and pg, and add a union entity, *e.g.* student, that represents the union set of both undergraduate and postgraduate students. This is illustrated in Figure 2, where the complete integrated schema S_{12} is presented.

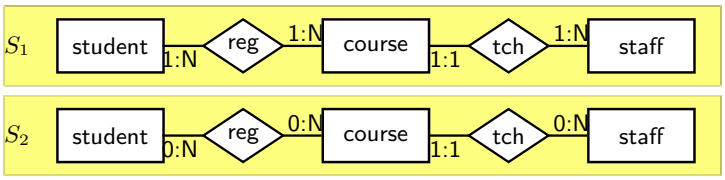


Fig. 1. Schema S_1 and S_2: undergraduate and postgraduate data sources

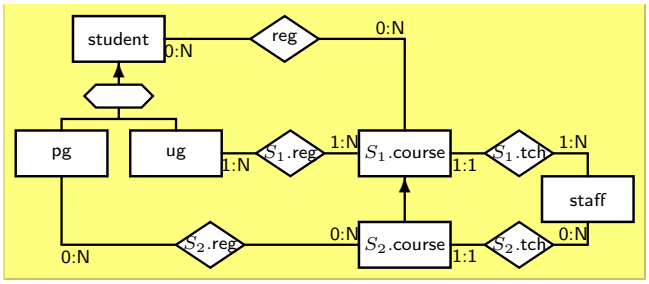


Fig. 2. Schema S_{12}: integration of S_1 and S_2

However, in general it is impossible to identify fully automatically the correct semantic mappings. Even in the small example above, where the schemas are almost identical, the semantics of the schema objects show subtle differences which make the discovery of the actual semantic mappings very difficult. Most existing techniques [9–11] try to identify a single mapping for each pair of objects, which of course could be wrong. For example, an automatic schema matching technique

might produce an **equivalence** mapping between the two student entities in S_1 and S_2, based on name comparison.

In this paper, we extend the concept of semantic mapping to include the notion of uncertainty, thus enabling schema matching techniques to show their level of belief on the mappings that they produce. Our goal is the management of this uncertainty. We do not include the implementation details of discovering uncertain mappings, nor the merging technique used to produce the integrated schema, even though we give such examples to illustrate our approach. To gain an intuition of our methodology, assume to have a finite amount of belief that can be distributed to the alternative semantic mappings of two schema objects. When we are certain about a mapping we assign all our belief to it. This is implicitly done by the existing schema matching techniques [1]. A straightforward extension of this concept can be obtained by allowing several alternative mappings to be possible, and distributing our belief to them. For example we might think that the two student entities are either disjoint (if we believe that one entity is undergraduates and the other postgraduates), or equivalent (if both entities represent all the students). This legitimate uncertainty should not prevent the integration of schemas S_1 and S_2. In fact we can think of two possible integrations, one based on disjointness, where one would form a generalisation hierarchy under student, as shown in Figure 2, and the other based on equivalence, where there would be just a single student entity in the final schema. Hence the uncertainty in the mapping between the two student entities propagates to the corresponding alternative integrations. The final integrated schema is created by combining all the produced mappings and it is structurally uncertain.

Our approach, which produces a set of possible mappings for each pair of schema objects, subsumes previous work where a single mapping is specified for each pair. As far as we know, there are two other related approaches that are concerned with uncertainty in schema and data integration. In [12] an approach to integrating XML documents is described, based on probability theory, that deals with uncertainty in data-level schemas. However, we focus on schema integration, and probability theory is just a particular case of the formalism used in our approach to manage uncertainty. In [13], uncertainty is only examined on equivalence mappings, while we provide a much wider set of possible semantic mappings, *e.g.* subsumption and disjointness. Moreover, in [13] only mappings between sets of attributes are considered, while we propose a more general methodology for matching and merging whole schemas.

The paper is organized as follows. In the next section we briefly present an existing schema integration method based on semantic mappings [14, 8]. In the following sections, we extend it to deal with uncertainty. In particular, in Section 3 we introduce the theory used to represent uncertainty, and provide the formal definition of *uncertain semantic relationship* (USR), together with illustrative examples. In the same section, we also present a software architecture that can be used to compare schemas and discover USRs. In Section 4 we analyze dependencies between USRs, and describe the process of building an uncertain integrated schema, *i.e.* a set of possible schemas with a belief distribution over

them. The main problem tackled in this section is the management of dependencies between USRs. Finally, we draw our concluding remarks. Schemas S_1 and S_2 in Figure 1 will be used as a working example throughout the paper.

2 Schema Integration based on Semantic Mappings

In this section, we summarize the schema integration approach presented in [14, 8], which we then extend in the sections that follow to deal with uncertainty.

2.1 Semantic Relationships

In [14], a mapping between two schema objects is specified by a semantic relationship. We have defined six types of semantic relationships between schema objects based on a set comparison of their *intentional domains*, *i.e.* the set of real-world objects that they represent [14]. We use $Dom_{int}(E)$ to define the intentional domain of an ER entity E. The intentional domain of a binary ER relationship is a subset of the Cartesian domain of the intentional domains of the entities it associates, *e.g.* in schema S_1, $Dom_{int}(\mathsf{reg}) \subseteq Dom_{int}(\mathsf{student}) \times Dom_{int}(\mathsf{course})$. The semantic relationships are:

1. **equivalence ($\overset{\mathbf{S}}{\equiv}$)**: Schema object ER_1 is equivalent to ER_2, $ER_1 \overset{S}{\equiv} ER_2$, iff $Dom_{int}(ER_1) = Dom_{int}(ER_2)$
2. **subset-subsumption ($\overset{\mathbf{S}}{\subset}$)**: Schema object ER_1 is a subset of schema object ER_2, $ER_1 \overset{S}{\subset} ER_2$, iff $Dom_{int}(ER_1) \subset Dom_{int}(ER_2)$
3. **superset-subsumption ($\overset{\mathbf{S}}{\supset}$)**: Schema object ER_1 is a superset of schema object ER_2, $ER_1 \overset{S}{\supset} ER_2$, iff $Dom_{int}(ER_1) \supset Dom_{int}(ER_2)$
4. **intersection ($\overset{\mathbf{S}}{\cap}$)**: Two schema objects ER_1 and ER_2 are intersecting, $ER_1 \overset{S}{\cap} ER_2$, iff $\neg(ER_1 \overset{S}{\subset} ER_2), \neg(ER_1 \overset{S}{\supset} ER_2), Dom_{int}(ER_1) \cap Dom_{int}(ER_2) \neq \emptyset, \exists ER_3 : Dom_{int}(ER_1) \cap Dom_{int}(ER_2) = Dom_{int}(ER_3)$
5. **disjointness ($\overset{\mathbf{S}}{\sqcap}$)**: Two schema objects ER_1 and ER_2 are disjoint, $ER_1 \overset{S}{\sqcap} ER_2$, iff $Dom_{int}(ER_1) \cap Dom_{int}(ER_2) = \emptyset, \exists ER_3 : Dom_{int}(ER_1) \cup Dom_{int}(ER_2) \subseteq Dom_{int}(ER_3)$
6. **incompatibility ($\overset{\mathbf{S}}{\not\sim}$)**: Two schema objects ER_1 and ER_2 are incompatible, $ER_1 \overset{S}{\not\sim} ER_2$, iff $Dom_{int}(ER_1) \cap Dom_{int}(ER_2) = \emptyset, \nexists ER_3 : Dom_{int}(ER_1) \cup Dom_{int}(ER_2) \subseteq Dom_{int}(ER_3)$

It is important to notice that object ER_3 in the definition of intersection and disjointness may or may not exist in the schemas. The notation $\exists ER_3 :$ *condition* means that there is a real-world concept in the domain of the data sources examined, that can be represented by an existing or non-existing schema object ER_3 that satisfies the *condition*. The notation $\nexists ER_3 :$ *condition* in the definition of incompatibility means that there is no real-world concept that would be represented by a schema object ER_3 to satisfy the specified *condition*. We term **semantically compatible** any two schema objects related by one of the above semantic relationships, except incompatibility.

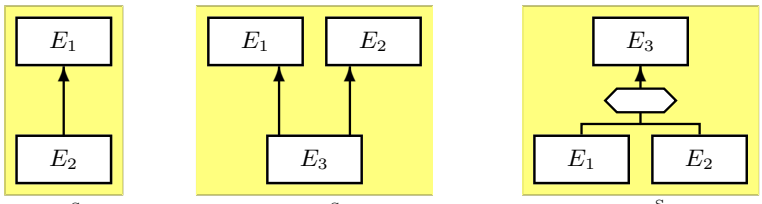
During schema matching, the identification of the above semantic relationships is accomplished by a bidirectional comparison. Our architecture consists of a pool of experts that exploit different types of information to compare schema objects, *e.g.* schema object names, cardinalities, instances, statistical data over the instances, data types, value ranges and lengths. The experts produce similarity degrees which are then aggregated, and with the help of user-defined thresholds the semantic relationships between the schema objects are specified. For example, the comparison of schemas S_1 and S_2 in Figure 1 could produce the following semantic mappings:

$S_1.\mathsf{student} \overset{S}{\cap} S_2.\mathsf{student}$ $S_1.\mathsf{course} \overset{S}{\not\sim} S_2.\mathsf{staff}$ $S_1.\mathsf{reg} \overset{S}{\cap} S_2.\mathsf{reg}$
$S_1.\mathsf{student} \overset{S}{\not\sim} S_2.\mathsf{course}$ $S_1.\mathsf{staff} \overset{S}{=} S_2.\mathsf{staff}$ $S_1.\mathsf{reg} \overset{S}{\not\sim} S_2.\mathsf{tch}$
$S_1.\mathsf{student} \overset{S}{\not\sim} S_2.\mathsf{staff}$ $S_1.\mathsf{staff} \overset{S}{\not\sim} S_2.\mathsf{student}$ $S_1.\mathsf{tch} \overset{S}{\not\sim} S_2.\mathsf{reg}$
$S_1.\mathsf{course} \overset{S}{\subset} S_2.\mathsf{course}$ $S_1.\mathsf{staff} \overset{S}{\not\sim} S_2.\mathsf{course}$ $S_1.\mathsf{tch} \overset{S}{\not\sim} S_2.\mathsf{tch}$
$S_1.\mathsf{course} \overset{S}{\not\sim} S_2.\mathsf{student}$

The generation of schema S_{12} in Figure 2 is based on these mappings. However, this 'definite' answer misses the fact that we may not be certain that some of the above mappings are correct, and hence alternative integrated schemas exist.

2.2 Schema Merging

In [8], we have defined the merging of schemas based on the semantic mappings specified between their schema objects. Formal rules have been defined that generate both-as-view (BAV) schema transformations [15] and merge two schemas. The application of three such rules on entities E_1 and E_2 is illustrated in Figure 3.



(a) $E_2 \overset{S}{\subset} E_1$ (b) $E_1 \overset{S}{\cap} E_2$ (c) $E_1 \overset{S}{\not\cap} E_2$

Fig. 3. Partial Integrated Schemas: ER Entity Subsumption, Intersection, Disjointness

Figure 3(a) illustrates the *partial integrated schema* that is created when a subsumption relationship is identified between two ER entities, *e.g.* the superset-subsumption relationship identified between entity course in S_1 and course in S_2. We call it a partial integrated schema because it is just a part of the final integrated schema. Figure 3(b) illustrates the partial integrated schema that is created when an intersection relationship is identified between two entities, and Figure 3(c) shows the partial integrated schema created when a disjointness relationship is identified between two entities, *e.g.* the two student entities in S_1 and S_2.

3 Uncertain Semantic Relationships

As already discussed in the introduction, an uncertain semantic mapping is a distribution of beliefs over the set of all possible semantic relationships. To represent beliefs, we have adopted Shafer's belief functions [16]. This choice is justified by the fact that Shafer's belief functions can represent the main kinds of uncertainty present in schema matching (as illustrated in the Examples 1–5 that follow).

The basic concept of Shafer's theory is a function called *basic probability assignment* (BPA), that assigns some probability mass to possible events. The set of all possible elementary events is called *frame of discernment*, and is represented by the letter Θ. In our case, Θ is the set of semantic relationships defined in Section 2, *i.e.* $\{\overset{\text{S}}{=}, \overset{\text{S}}{\cap}, \overset{\text{S}}{\subset}, \overset{\text{S}}{\supset}, \overset{\text{S}}{\not\cap}, \overset{\text{S}}{\not\sim}\}$. Possible events correspond to subsets of Θ. For instance, the set $\{\overset{\text{S}}{=}, \overset{\text{S}}{\cap}\}$ represents the event "The correct semantic relationship is either equivalence or intersection", and $m(\{\overset{\text{S}}{=}, \overset{\text{S}}{\cap}\})$ is the probability mass supporting exactly this event.

Definition 1 (Basic Probability Assignment (BPA)). *A function* $m : 2^{\Theta} \rightarrow [0, 1]$ *is called* basic probability assignment *whenever:*

- $m(\emptyset) = 0$
- $\sum_{A \subseteq \Theta} m(A) = 1$

From a BPA function, we can compute the belief and plausibility of any subset A of Θ.

$$\text{Bel}(A) = \sum_{B \subseteq A} m(B) \tag{1}$$

$$\text{Pl}(A) = \sum_{B \subseteq \Theta, B \cap A \neq \emptyset} m(B) \tag{2}$$

Belief in A is the sum of all probability masses assigned to subsets of A. For example, let A be the set $\{\overset{\text{S}}{=}, \overset{\text{S}}{\not\sim}\}$. If we assign some probability mass to the set $\{\overset{\text{S}}{=}\}$, this increases our belief in all the events containing it. In fact, if we have some evidence supporting the event "The true semantic relationship is equivalence", the same evidence increases also our belief in the event "The true semantic relationship is either equivalence or incompatibility". Plausibility of $A = \{\overset{\text{S}}{=}, \overset{\text{S}}{\not\sim}\}$ is the sum of all probability masses that are compatible with $\{\overset{\text{S}}{=}, \overset{\text{S}}{\not\sim}\}$. For example, some probability mass assigned to $\{\overset{\text{S}}{=}, \overset{\text{S}}{\not\cap}\}$ tells us that A is plausible, without increasing our belief in it, because the right relationship could be disjointness. These definitions can be used to formally define an USR:

Definition 2 (Uncertain Semantic Relationship (USR)). *An* uncertain semantic relationship *between two schema objects* A *and* B *is a pair* (Θ, m)*, where* $\Theta = \{\overset{S}{=}, \overset{S}{\cap}, \overset{S}{\subset}, \overset{S}{\supset}, \overset{S}{\not\cap}, \overset{S}{\not\sim}\}$ *and* m *is a BPA.*

In the following examples we present the main possible types of USRs, to show that Shafer's theory is expressive enough to represent all USRs that can be found in schema integration.

Example 1 (Certain Relationship). A certain semantic relationship is a special case of USR, where all the probability mass is assigned to a singleton. For example, a BPA $m(\{\overset{S}{=}\}) = 1$ means that we are sure that the true relationship is equivalence.

Example 2 (Probabilistic Relationship). We can use m to assign probabilities to alternative relationships. A BPA $m(\{\overset{S}{\not\cap}\}) = .4, m(\{\overset{S}{\not\sim}\}) = .6$ means that the probability of disjointness is .4, while the probability of incompatibility is .6.

Example 3 (Non-specific Relationship). In many cases, we will only be able to restrict Θ, *i.e.* to exclude some relationships. If we know that two objects are not equivalent, and that the first cannot be a subset of the second, the corresponding BPA will be $m(\{\overset{S}{\cap}, \overset{S}{\supset}, \overset{S}{\not\cap}, \overset{S}{\not\sim}\}) = 1$.

Example 4 (Partial Ignorance). When we have some information supporting one or more relationships, we should commit part of our belief to them. For instance, a BPA $m(\{\overset{S}{=}\}) = .2, m(\{\Theta\}) = .8$ means that we have some evidence that two objects are equivalent, but we are not sure. Notice that in this case m does not define probabilities. A typical problem with probabilities is the difficulty to justify their precise numerical values. The BPA presented in this example is much more flexible, as it corresponds to a belief $\text{Bel}(\{\overset{S}{=}\}) = .2$ and a plausibility $\text{Pl}(\{\overset{S}{=}\}) = 1$, and thus defines a confidence interval $[.2, 1]$ on the equivalence relationship.

Example 5 (Total Ignorance). As a final example, consider a case in which we have no information about two objects, or we do not want to compare them. This can be very useful to compare parts of schemas, as we show in Section 3.1. We can express our ignorance using the following BPA: $m(\{\Theta\}) = 1$.

3.1 Discovery of USRs

The concept of USR defined above is very intuitive, and is supported by a well known theory at the same time. In this section we present an architecture to discover USRs, and provide an example of schema matching between two entities of S_1 and S_2.

As in the method described in Section 2, the comparison of schema objects is performed by a pool of experts, each one specialized on some features. However, to support the inherent uncertainty of schema matching, experts produce USRs. The mapping between any two schema objects is computed by aggregating the results of all the available experts. Our architecture is illustrated in Figure 4.

The aggregation of USRs is easily achieved by using Dempster's combination rule, that takes two BPAs over the same frame of discernment Θ as input [16]. Using this rule, the combination of experts' beliefs is both based on a sound

theory and easy to implement. For every subset A of Θ, the combination of two beliefs (defined by BPAs m_1 and m_2) is defined as:

$$m(A) = \begin{cases} 0 & \text{if } A = \emptyset \\ \frac{\sum_{A_1 \subseteq \Theta, A_2 \subseteq \Theta, A_1 \cap A_2 = A} m_1(A_1) m_2(A_2)}{1 - \sum_{A_1 \subseteq \Theta, A_2 \subseteq \Theta, A_1 \cap A_2 = \emptyset} m_1(A_1) m_2(A_2)} & \text{if } A \neq \emptyset \end{cases} \quad (3)$$

This rule can be used to combine the USRs produced by two experts. The combination of the beliefs of n experts is obtained by iteratively applying it $n-1$ times.

After the application of the rule, it may happen that some semantic relationships are supported by a very small amount of probability mass. In this case, we can decide to dispose of them, and to keep only the relationships supported by a significant amount of probability mass. Thresholds can be used for this purpose. This is useful to improve efficiency, as it reduces the cardinality of the event space, and it allows us not to consider possible semantic relationships that are very unlikely to be the correct ones. However, in this paper we do not investigate how to choose thresholds, as we focus on the theoretical aspects of our method. In general, they can be found experimentally, or set up by the users.

Our architecture has many desirable features: (a) its implementation can be focused on experts, that can be very small independent software agents, (b) it is scalable, as experts can be deleted and added to the pool with no complexity, (c) it is easily parallelizable, as experts can run on different and dedicated hardware, and (d) experts can be software modules, equipped with data analysis tools, or they can be humans, using software interfaces.

The only requirement on experts is to output USRs. Dempster's rule can be used as long as they do not contradict each other. Therefore, human experts can cooperate with software agents to improve the quality of integration of large schemas, thanks to Dempster's combination rule. If a human expert knows or identifies with certainty some relationships, the beliefs of other experts will not be considered, as far as they do not state explicitly that the human USR is wrong. At the same time, we can expect human experts to give their contribution on some parts of the schemas, letting software agents compare the remaining schema objects. This can be done by expressing total ignorance about the objects we do not want to compare. Total ignorance does not influence the combination of beliefs of other experts.

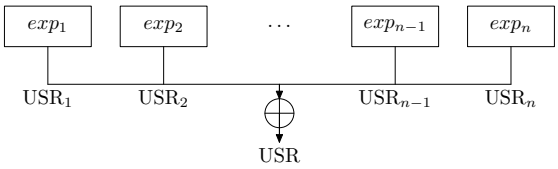


Fig. 4. Architecture proposed to discover USRs.

3.2 Examples

To clarify how USRs can be discovered, we present an example involving three experts. However, the definition of experts lies outside the scope of this paper, and we introduce them only to show how our architecture works. This example focuses on the comparison of the two student entities in schemas S_1 and S_2.

The first expert compares the cardinality of two schema objects, *i.e.* the number of instances belonging to them. If cardinalities are equal, subsumption is not possible. If the cardinality of the first object is greater than the other, they cannot be equivalent and the second object cannot subsume the first one. Notice that this expert assumes that all instances belonging to those objects in the real world are stored in the database. It would be easy to improve the expert so that some instances can be missing, using fuzzy comparisons. However, this is not needed in this example. The cardinality of S_1.student is much greater than that of S_2.student, because there are much more undergraduate students than postgraduates. Therefore, the first expert can exclude equivalence and subset-subsumption. The USR produced by this expert is defined by $m_1(\{\overset{S}{\cap}, \overset{S}{\supset}, \overset{S}{\not\cap}, \overset{S}{\not\supset}\}) = 1$.

The second expert compares object names, using an ontology. The ontology stores information about the six relationships under consideration, when comparing English words. For example, a subsumption relationship between the terms *undergraduate* and *student* corresponds to some confidence on the fact that a schema object whose name is undergraduate is a subset of a schema object called student. Moreover, the expert would also have some (less) confidence about the equality of the two corresponding objects. As it only compares the names of the entities, the ontology-based expert will always assume to be possibly wrong. The second expert, based on the identical names of student entities, might produce the following BPA: $m_2(\{\overset{S}{=}\}) = .7, m_2(\{\overset{S}{\subset}, \overset{S}{\supset}, \overset{S}{\not\cap}, \overset{S}{\cap}\}) = .2, m_2(\{\Theta\}) = .1$.

The third expert compares the instances of two schema objects. For efficiency reasons, it only compares a subset of the instances of S_1.student with all the instances of the S_2.student entity, and *vice versa*. Obviously, the expert cannot compare directly real-world objects, but must compare name, type, and values of the entity identifiers in the ER schemas. This induces uncertainty on the result. In our example, the third expert cannot find matches between the instances of the two student entities, because an undergraduate cannot be a postgraduate and *vice versa*. Therefore, it will support the set of relationships $\{\overset{S}{\cap}, \overset{S}{\not\cap}\}$. However, as already said, the expert cannot be certain of this information. Its USR is defined by: $m_3(\{\overset{S}{\cap}, \overset{S}{\not\cap}\}) = .8, m_3(\{\Theta\}) = .2$.

The combination of m_1, m_2, and m_3 is obtained by applying Dempster's rule, and produces the following USR:

$$m(\{\overset{S}{\cap}, \overset{S}{\not\cap}\}) = 4/5,\ m(\{\overset{S}{\cap}, \overset{S}{\supset}, \overset{S}{\not\cap}\}) = 2/15,\ m(\{\overset{S}{\cap}, \overset{S}{\supset}, \overset{S}{\not\cap}, \overset{S}{\not\supset}\}) = 1/15\ .$$

This result is what we would expect from the combination of the three USRs. The second expert assigns a large amount of probability mass to the equivalence relationship, but this option is excluded by the first expert. For this reason,

equivalence is not considered in the final USR. The high probability mass assigned to disjointness and intersection is justified by the fact that m_1, m_2 and m_3 support these two relationships. In fact, all experts think that disjointness and intersection are plausible, and one of them (the third expert) believes in it.

Relationship	Bel	Pl
$\{\overset{S}{=}\}$	0	0
$\{\overset{S}{\subset}\}$	0	0
$\{\overset{S}{\supset}\}$	0	$\frac{1}{5}$
$\{\overset{S}{\cap}\}$	0	1
$\{\overset{S}{\not\cap}\}$	0	1
$\{\overset{S}{\not\sim}\}$	0	$\frac{1}{15}$

Table 1. Belief, plausibility of alternative semantic relationships between students

In Table 1 we have indicated belief and plausibility of every alternative relationship. Notice that both $\overset{S}{\cap}$ and $\overset{S}{\not\cap}$ are completely plausible, while $\overset{S}{=}$ and $\overset{S}{\subset}$ are not plausible at all. The choice of further considering $\overset{S}{\supset}$ and $\overset{S}{\not\sim}$ in our analysis depends on the threshold we set up. In our working example, we will not consider them as they are not plausible enough, compared to $\overset{S}{\cap}$ and $\overset{S}{\not\cap}$.

4 Uncertain Integrated Schema

This section presents the schema merging process of our methodology. Based on schema matching and the discovered uncertain semantic relationships, several possible integrated schemas can be created. We explain how the beliefs assigned to the uncertain semantic relationships are propagated to these schemas and a final *uncertain integrated schema* is produced. First, though, the dependencies between the uncertain semantic relationships need to be examined and possible conflicts need to be identified.

4.1 Dependencies between Semantic Relationships

Consider again the two schemas S_1 and S_2. Similarly to Section 3.2, the uncertain semantic relationships between the two reg ER relationships can be computed. These two ER relationships have identical names but they do not have any instances in common and particularly S_1.reg represents a much larger set of instances. Thus, the three experts described in Section 3.2 will produce the same USRs as the ones produced for students. These are aggregated and the highest probability mass is assigned to disjointness and intersection, $m(\{\overset{S}{\cap}, \overset{S}{\not\cap}\}) = \frac{4}{5}$. Because the rest of the alternatives have very small probability masses we can safely assume that $m(\{\overset{S}{\cap}, \overset{S}{\not\cap}\}) = 1$. The same assumption will also produce $m(\{\overset{S}{\cap}, \overset{S}{\not\cap}\}) = 1$ for the student entities. Finally, suppose that a human expert has

specified that the semantic relationship between the course entities is superset-subsumption, $m(\{\overset{S}{\supset}\}) = 1$, *i.e.* $S_1.\mathsf{course} \overset{S}{\supset} S_2.\mathsf{course}$.

During schema merging, these produced USRs need to be combined. Table 2 illustrates all their possible combinations. Consider the second row of the table, where $S_1.\mathsf{course} \overset{S}{\supset} S_2.\mathsf{course}$, $S_1.\mathsf{reg} \overset{S}{\cap} S_2.\mathsf{reg}$ and $S_1.\mathsf{student} \overset{S}{\not\cap} S_2.\mathsf{student}$. The intersection relationship between the two reg ER relationships specifies that there is at least one common instance between $S_1.\mathsf{reg}$ and $S_2.\mathsf{reg}$, *i.e.* there is a common instance of $S_1.\mathsf{student}$ and $S_2.\mathsf{student}$ that is associated with a common instance of $S_1.\mathsf{course}$ and $S_2.\mathsf{course}$. But, according to the second row of the table, the student entities are disjoint and do not have any instances in common. Therefore, the combination of semantic relationships in the second row of the table is invalid.

S_1.course,S_2.course	S_1.reg,S_2.reg	S_1.student,S_2.student
$\overset{S}{\supset}$	$\overset{S}{\cap}$	$\overset{S}{\cap}$
$\overset{S}{\supset}$	$\overset{S}{\cap}$	$\overset{S}{\not\cap}$
$\overset{S}{\supset}$	$\overset{S}{\not\cap}$	$\overset{S}{\cap}$
$\overset{S}{\supset}$	$\overset{S}{\not\cap}$	$\overset{S}{\not\cap}$

Table 2. Possible combinations of alternative semantic relationships between course, reg, and student schema objects

This example manifests the existence of *dependencies* between the semantic relationships of ER relationships and the semantic relationships of the associated ER entities, and *vice versa*. In this paper, we focus just on binary ER relationships. We have exhaustively examined their dependencies and we present in Table 3 all the legal combinations.

The table considers the general case of two ER relationships: ER relationship R_1 that associates ER entities A_1 and B_1 and ER relationship R_2 that associates entities A_2 and B_2 (Figure 5). The first column of the table specifies the semantic relationship between the entities A_1 and A_2, and the second column specifies the semantic relationship between B_1 and B_2. The third column examines the possible semantic relationships between R_1 and R_2.

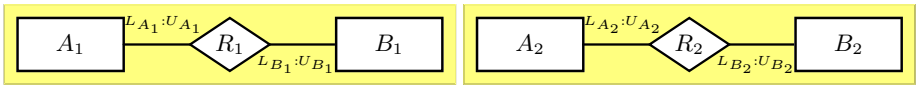


Fig. 5. Two ER relationships: R_1 and R_2

Our previous example, where the intersection relationship between $S_1.\mathsf{reg}$ and $S_2.\mathsf{reg}$ was invalid, is a case of $A_1 \overset{S}{\subset} A_2$, $B_1 \overset{S}{\not\cap} B_2$ instantiated to $S_2.\mathsf{course} \overset{S}{\subset} S_1.\mathsf{course}$, $S_2.\mathsf{student} \overset{S}{\not\cap} S_1.\mathsf{student}$. Row nine of Table 3 defines that in this case the legal semantic relationships between R_1 and R_2, instantiated to $S_2.\mathsf{reg}$ and $S_1.\mathsf{reg}$,

A_1,A_2	B_1,B_2	R_1,R_2
$\overset{s}{=}$	$\overset{s}{=}$	$\overset{s}{\not\sim}$, $\overset{s}{\not\cap}$, $\overset{s}{\cap}$: always possible, $\overset{s}{=}$: L_{A_1}=L_{A_2}, U_{A_1}=U_{A_2}, L_{B_1}=L_{B_2}, U_{B_1}=U_{B_2}, $\overset{s}{\subset}$: $L_{A_1} \leq L_{A_2}$, $U_{A_1} \leq U_{A_2}$, $\neg$(L_{A_1}=L_{A_2}, U_{A_1}=U_{A_2}=(1,1)), $L_{B_1} \leq L_{B_2}$, $U_{B_1} \leq U_{B_2}$, $\neg$(L_{B_1}=L_{B_2}, U_{B_1}=U_{B_2}=(1,1)), $\overset{s}{\supset}$: $L_{A_1} \geq L_{A_2}$, $U_{A_1} \geq U_{A_2}$, $\neg$(L_{A_1}=L_{A_2}, U_{A_1}=U_{A_2}=(1,1)), $L_{B_1} \geq L_{B_2}$, $U_{B_1} \geq U_{B_2}$, $\neg$(L_{B_1}=L_{B_2}, U_{B_1}=U_{B_2}=(1,1)).
$\overset{s}{=}$	$\overset{s}{\subset}$	$\overset{s}{\not\sim}$, $\overset{s}{\not\cap}$, $\overset{s}{\cap}$: always possible, $\overset{s}{=}$: L_{A_1}=L_{A_2}, U_{A_1}=U_{A_2}, U_{B_1}=U_{B_2}, L_{B_1}=0, $\overset{s}{\subset}$: $L_{A_1} \leq L_{A_2}$, $U_{A_1} \leq U_{A_2}$, $U_{B_1} \leq U_{B_2}$, $\overset{s}{\supset}$: $L_{A_1} \geq L_{A_2}$, $U_{A_1} \geq U_{A_2}$, L_{B_2}=0, $U_{B_1} \geq U_{B_2}$.
$\overset{s}{=}$	$\overset{s}{\subset}$	$\overset{s}{\not\sim}$, $\overset{s}{\not\cap}$, $\overset{s}{\cap}$: always possible, $\overset{s}{=}$: L_{A_1}=L_{A_2}, U_{A_1}=U_{A_2}, L_{B_1}=L_{B_2}=0, $\overset{s}{\subset}$: $L_{A_1} \leq L_{A_2}$, $U_{A_1} \leq U_{A_2}$, $\neg$(L_{A_1}=L_{A_2}=1, U_{A_1}=U_{A_2}=1), L_{B_1}=0, $U_{B_1} \leq U_{B_2}$, $\overset{s}{\supset}$: $L_{A_1} \geq L_{A_2}$, $U_{A_1} \geq U_{A_2}$, $\neg$(L_{A_1}=L_{A_2}=1, U_{A_1}=U_{A_2}=1), L_{B_2}=0, $U_{B_1} \geq U_{B_2}$.
$\overset{s}{=}$	$\overset{s}{\not\cap}$	$\overset{s}{\not\sim}$, $\overset{s}{\not\cap}$: always possible.
$\overset{s}{=}$	$\overset{s}{\not\sim}$	$\overset{s}{\not\sim}$: always possible.
$\overset{s}{\subset}$	$\overset{s}{\subset}$	$\overset{s}{\not\sim}$, $\overset{s}{\not\cap}$, $\overset{s}{\cap}$: always possible, $\overset{s}{\subset}$: $U_{A_1} \leq U_{A_2}$, $U_{B_1} \leq U_{B_2}$, $\overset{s}{\supset}$: $U_{A_1} \geq U_{A_2}$, $U_{B_1} \geq U_{B_2}$, L_{A_2}=L_{B_2}=0, $\overset{s}{=}$: U_{A_1}=U_{A_2}, U_{B_1}=U_{B_2}, L_{A_2}=L_{B_2}=0.
$\overset{s}{\subset}$	$\overset{s}{\supset}$	$\overset{s}{\not\sim}$, $\overset{s}{\not\cap}$, $\overset{s}{\cap}$: always possible, $\overset{s}{\subset}$: L_{B_1}=0, $U_{B_1} \leq U_{B_2}$, $U_{A_1} \leq U_{A_2}$, $\overset{s}{\supset}$: $U_{A_1} \geq U_{A_2}$, $U_{B_1} \geq U_{B_2}$, L_{A_2}=0, $\overset{s}{=}$: U_{A_1}=U_{A_2}, U_{B_1}=U_{B_2}, L_{A_2}=L_{B_1}=0.
$\overset{s}{\subset}$	$\overset{s}{\subset}$	$\overset{s}{\not\sim}$, $\overset{s}{\not\cap}$, $\overset{s}{\cap}$: always possible, $\overset{s}{\subset}$: L_{B_1}=0, $U_{B_1} \leq U_{B_2}$, $U_{A_1} \leq U_{A_2}$, $\overset{s}{\supset}$: $U_{A_1} \geq U_{A_2}$, $U_{B_1} \geq U_{B_2}$, L_{A_2}=L_{B_2}=0, $\overset{s}{=}$: U_{A_1}=U_{A_2}, U_{B_1}=U_{B_2}, L_{B_1}=L_{B_2}=L_{A_2}=0.
$\overset{s}{\subset}$	$\overset{s}{\not\cap}$	$\overset{s}{\not\sim}$, $\overset{s}{\not\cap}$: always possible.
$\overset{s}{\subset}$	$\overset{s}{\not\sim}$	$\overset{s}{\not\sim}$: always possible.
$\overset{s}{\subset}$	$\overset{s}{\subset}$	$\overset{s}{\not\sim}$, $\overset{s}{\not\cap}$, $\overset{s}{\cap}$: always possible, $\overset{s}{\subset}$: L_{A_1}=0, $U_{A_1} \leq U_{A_2}$, L_{B_1}=0, $U_{B_1} \leq U_{B_2}$, $\overset{s}{\supset}$: L_{A_2}=0, $U_{A_2} \leq U_{A_1}$, L_{B_2}=0, $U_{B_2} \leq U_{B_1}$, $\overset{s}{=}$: L_{A_1}=L_{A_2}=0,L_{B_1}=L_{B_2}=0, U_{A_1}=U_{A_2}, U_{B_1}=U_{B_2}.
$\overset{s}{\subset}$	$\overset{s}{\not\cap}$	$\overset{s}{\not\sim}$, $\overset{s}{\not\cap}$: always possible.
$\overset{s}{\subset}$	$\overset{s}{\not\sim}$	$\overset{s}{\not\sim}$: always possible.
$\overset{s}{\not\cap}$	$\overset{s}{\not\cap}$	$\overset{s}{\not\sim}$, $\overset{s}{\not\cap}$: always possible.
$\overset{s}{\not\cap}$	$\overset{s}{\not\sim}$	$\overset{s}{\not\sim}$: always possible.
$\overset{s}{\not\sim}$	$\overset{s}{\not\sim}$	$\overset{s}{\not\sim}$: always possible.

Table 3. Dependencies between entities and ER relationships

are only incompatibility and disjointness. Thus, the intersection relationship between them is invalid, as previously shown.

In some cases, a semantic relationship between R_1 and R_2 can only be legal when a cardinality condition is satisfied, *e.g.* we can have that $A_1 \overset{S}{=} A_2$, $A_2 \overset{S}{=} B_2$, $R_1 \overset{S}{=} R_2$ if and only if the cardinalities of R_1 and R_2 are identical (first row of Table 3).

A,B	B,C	A,C
$\overset{S}{=}$	$\overset{S}{\subset}$	$\overset{S}{\subset}$
$\overset{S}{=}$	$\overset{S}{\supset}$	$\overset{S}{\supset}$
$\overset{S}{=}$	$\overset{S}{\not\sim}$	$\overset{S}{\not\sim}$
$\overset{S}{\subset}$	$\overset{S}{\supset}$	$\overset{S}{=}$, $\overset{S}{\cap}$, $\overset{S}{\subset}$, $\overset{S}{\supset}$, $\overset{S}{\not\supset}$, $\overset{S}{\not\sim}$
$\overset{S}{\subset}$	$\overset{S}{\not\supset}$	$\overset{S}{\not\supset}$, $\overset{S}{\not\sim}$
$\overset{S}{\supset}$	$\overset{S}{\supset}$	$\overset{S}{\supset}$
$\overset{S}{\supset}$	$\overset{S}{\not\supset}$	$\overset{S}{\cap}$, $\overset{S}{\supset}$, $\overset{S}{\not\supset}$, $\overset{S}{\not\sim}$
$\overset{S}{\cap}$	$\overset{S}{\supset}$	$\overset{S}{=}$, $\overset{S}{\cap}$, $\overset{S}{\subset}$, $\overset{S}{\supset}$, $\overset{S}{\not\supset}$, $\overset{S}{\not\sim}$
$\overset{S}{\cap}$	$\overset{S}{\not\sim}$	$\overset{S}{\cap}$, $\overset{S}{\supset}$, $\overset{S}{\not\supset}$, $\overset{S}{\not\sim}$
$\overset{S}{\not\supset}$	$\overset{S}{\not\sim}$	$\overset{S}{\cap}$, $\overset{S}{\subset}$, $\overset{S}{\supset}$, $\overset{S}{\not\supset}$, $\overset{S}{\not\sim}$

A,B	B,C	A,C
$\overset{S}{=}$	$\overset{S}{\supset}$	$\overset{S}{\supset}$
$\overset{S}{=}$	$\overset{S}{\not\supset}$	$\overset{S}{\not\supset}$
$\overset{S}{\subset}$	$\overset{S}{\subset}$	$\overset{S}{\subset}$
$\overset{S}{\subset}$	$\overset{S}{\cap}$	$\overset{S}{\cap}$, $\overset{S}{\subset}$, $\overset{S}{\not\supset}$, $\overset{S}{\not\sim}$
$\overset{S}{\subset}$	$\overset{S}{\not\sim}$	$\overset{S}{\not\supset}$, $\overset{S}{\not\sim}$
$\overset{S}{\supset}$	$\overset{S}{\cap}$	$\overset{S}{\supset}$, $\overset{S}{\cap}$
$\overset{S}{\supset}$	$\overset{S}{\not\sim}$	$\overset{S}{\cap}$, $\overset{S}{\supset}$, $\overset{S}{\not\supset}$, $\overset{S}{\not\sim}$
$\overset{S}{\cap}$	$\overset{S}{\not\supset}$	$\overset{S}{\cap}$, $\overset{S}{\supset}$, $\overset{S}{\not\supset}$, $\overset{S}{\not\sim}$
$\overset{S}{\not\supset}$	$\overset{S}{\not\supset}$	$\overset{S}{=}$, $\overset{S}{\cap}$, $\overset{S}{\subset}$, $\overset{S}{\supset}$, $\overset{S}{\not\supset}$, $\overset{S}{\not\sim}$
$\overset{S}{\not\sim}$	$\overset{S}{\not\sim}$	$\overset{S}{=}$, $\overset{S}{\cap}$, $\overset{S}{\subset}$, $\overset{S}{\supset}$, $\overset{S}{\not\supset}$, $\overset{S}{\not\sim}$

Table 4. Dependencies between schema objects of the same kind

Except from dependencies between the semantic relationships of ER relationships and the semantic relationships of their associated ER entities, there are also dependencies between the semantic relationships of the same type of constructs. Consider the following example. The ER relationship $S_1.\mathsf{tch}$ subsumes $S_2.\mathsf{tch}$ but we might be uncertain about the semantic relationship between $S_1.\mathsf{reg}$ and $S_2.\mathsf{tch}$ since both of them associate person identifiers with course identifiers. $S_1.\mathsf{reg}$ has a much larger set of instances than $S_2.\mathsf{tch}$, and therefore equivalence and subset-subsumption relationships are excluded. Thus, from a comparison of $S_1.\mathsf{reg}$ and $S_2.\mathsf{tch}$ a pool of experts could decide to support the set $\{\overset{S}{\supset}, \overset{S}{\cap}, \overset{S}{\not\sim}\}$ of possible semantic relationships. However, since $S_1.\mathsf{reg}$ and $S_1.\mathsf{tch}$ are incompatible, based on the structure of S_1, and $S_1.\mathsf{tch}$ subsumes $S_2.\mathsf{tch}$, the intersection and superset-subsumption relationships between $S_1.\mathsf{reg}$ and $S_2.\mathsf{tch}$ are also excluded. Therefore, $S_1.\mathsf{reg}$ and $S_2.\mathsf{tch}$ must be incompatible.

This restriction of relationships is generalised in Table 4, where all legal combinations of semantic relationships between three objects A, B and C of the same type of construct are defined. Objects B and C belong to the same schema thus their semantic relationship can be derived from the schema structure. Semantic relationships between A,B and A,C are discovered during schema matching. In our example of $S_1.\mathsf{reg}$ and $S_2.\mathsf{tch}$, A is instantiated to $S_2.\mathsf{tch}$ and B,C to $S_1.\mathsf{tch}$ and $S_1.\mathsf{reg}$, respectively. If the semantic relationships $S_2.\mathsf{tch} \overset{S}{\subset} S_1.\mathsf{tch}$ and $S_1.\mathsf{tch} \overset{S}{\not\sim} S_1.\mathsf{reg}$ are certain, then based on Table 4 $S_1.\mathsf{reg}$ and $S_2.\mathsf{tch}$ can only be disjoint or incompatible.

4.2 Schema Merging

#	S_1.stud.,S_2.stud.	S_1.reg,S_2.reg	S_1.course,S_2.course	S_1.staff,S_2.staff	S_1.tch,S_2.tch
(a)	$\overset{s}{\not\cap}$	$\overset{s}{\not\cap}$	$\overset{s}{\supset}$	$\overset{s}{\equiv}$	$\overset{s}{\supset}$
(b)	$\overset{s}{\not\cap}$	$\overset{s}{\not\cap}$	$\overset{s}{\supset}$	$\overset{s}{\supset}$	$\overset{s}{\supset}$
(c)	$\overset{s}{\cap}$	$\overset{s}{\cap}$	$\overset{s}{\supset}$	$\overset{s}{\equiv}$	$\overset{s}{\supset}$
(d)	$\overset{s}{\cap}$	$\overset{s}{\cap}$	$\overset{s}{\supset}$	$\overset{s}{\supset}$	$\overset{s}{\supset}$
(e)	$\overset{s}{\cap}$	$\overset{s}{\not\cap}$	$\overset{s}{\supset}$	$\overset{s}{\equiv}$	$\overset{s}{\supset}$
(f)	$\overset{s}{\cap}$	$\overset{s}{\not\cap}$	$\overset{s}{\supset}$	$\overset{s}{\supset}$	$\overset{s}{\supset}$

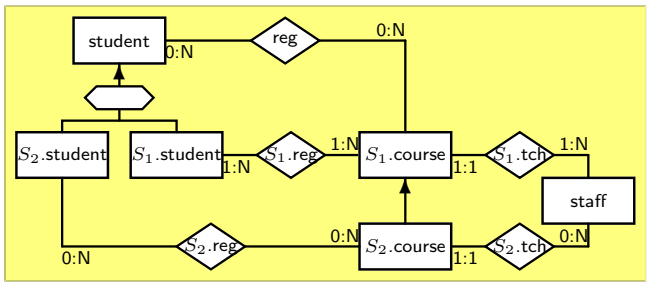
Table 5. Possible combinations of semantic relationships in the integrated schema

In the previous sections we compared student and reg schema objects, obtaining a set of possible semantic relationships between them, with BPAs representing our belief distribution. In particular, both student and reg schema objects could be either disjoint or intersecting. This is shown in Table 2.

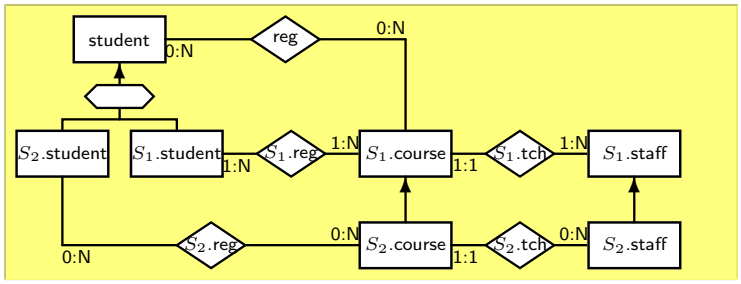
Now assume that S_1.course $\overset{s}{\supset}$ S_2.course and S_1.tch $\overset{s}{\supset}$ S_2.tch relationships are certain, while the relationship between S_1.staff and S_2.staff could be $\overset{s}{\equiv}$, with a probability of .7, or $\overset{s}{\supset}$, with a probability of .3. We can build a complete table (Table 5), that is an extension of Table 2, representing all possible combinations of semantic relationships between all pairs of schema objects. In Table 5 we have concentrated only on compatible objects. Each row of this final table corresponds to a possible integrated schema, where each semantic relationship defines a partial integrated schema, like those represented in Figure 3. For example, in the possible integrated schema (a) of Table 5 staff entities are equivalent, while in the possible integrated schema (b) S_1.staff subsumes S_2.staff. Based on this table we can create the corresponding schemas. The schemas corresponding to rows (a) and (b) of Table 5 are illustrated in Figure 6.

The BPA obtained as a combination of all the aforementioned USRs is defined by $m\{(a), (c), (e)\} = .7$, and $m\{(b), (d), (f)\} = .3$. The corresponding beliefs and plausibilities can be easily computed using (1) and (2). The meaning of this BPA reflects the uncertainty on the partial integrated schemas. The set {(a), (b), (c), (d), (e), (f)}, together with its BPA, is called an *uncertain integrated schema*, and is the final product of our schema integration approach on our working example.

From the uncertain integrated schema we can reconstruct all the previously produced USRs. For example, we previously assigned a probability mass of 1 to the set of relationships $\{\overset{s}{\equiv}, \overset{s}{\cap}\}$ between the two student entities. This value can be obtained from the uncertain integrated schema by adding together all the probability masses assigned to combinations of possible integrated schemas where S_1.student $\overset{s}{\not\cap}$ S_2.student or S_1.student $\overset{s}{\cap}$ S_2.student. This corresponds to all the rows of Table 5, *i.e.* all possible schemas. Similarly, if we sum all masses assigned to possible combinations of schemas where S_1.staff $\overset{s}{\equiv}$ S_2.staff, we obtain .7, while for S_1.staff $\overset{s}{\supset}$ S_2.staff we obtain .3.



(a)



(b)

Fig. 6. Two of the final alternative integrated schemas generated by our approach

5 Conclusion and Future Work

In this paper we have presented a new method of schema integration. Differently from other existing methods, our approach manages the inherent uncertainty in (semi-)automatic schema matching, and supports six kinds of semantic relationships between schema objects. These features are essential to cope with real schema integration tasks, where many semantic relationships are possible, and it is very unlikely to know all of them with certainty.

An analysis of the computational complexity of our method is outside the scope of this paper. However, it is easy to identify two main possible causes of inefficiency related to the management of uncertainty. The first is the combination of the USRs produced by the experts. In fact, the complexity of *exact methods* for performing Dempster's combination rule is exponential on the size of the frame of discernment, because it must consider all its subsets in the worst case. However, the frame of discernment in our method contains only six elements – our semantic relationships. Therefore, the complexity of the combination is bounded by a small constant. The second issue is the number of possible integrated schemas generated by the method, that can be exponential on the number of schema objects. However, in practice the output of our method will not be the set of all possible integrated schemas, but only the most probable ones. The number of schemas returned by the method can be decided in advance. Finally,

an appropriate use of thresholds can further reduce the number of schemas, without losing significant information.

While the theory underlying our method has been presented in this paper, we still need to experimentally verify its efficiency and effectiveness. In the future, we are going to implement it as an extension of an existing schema integration software [14].

References

1. Rahm, E., Bernstein, P.: A survey of approaches to automatic schema matching. VLDB Journal **10** (2001) 334–350
2. Bernstein, P.: Applying model management to classical meta data problems. In: Proc. CIDR (2003) 209–220
3. Batini, C., Lenzerini, M., Navathe, S.: A comparative analysis of methodologies for database schema integration. ACM Computing Surveys **18** (1986) 323–364
4. Bernstein, P.A., Pottinger, R.A.: Merging models based on given correspondences. In: Proc. 29th VLDB Conference, Berlin (2003)
5. Melnik, S., Rahm, E., Bernstein, P.A.: Rondo: a programming platform for generic model management. In: Proc. SIGMOD, ACM Press (2003) 193–204
6. Hayne S., Ram S.: Multi-User View Integration System (MUVIS): An Expert System for View Integration. In: ICDE (1990) 402–409
7. Spaccapietra, S., Parent, C.: View Integration: A Step Forward in Solving Structural Conflicts. IEEE TKDE, 6(2), (1994) 258–274
8. Rizopoulos, N., McBrien, P.: A general approach to the generation of conceptual model transformations. In: Proc. CAiSE. LNCS, Springer-Verlag (2005)
9. Madhavan, J., Bernstein, P., Rahm, E.: Generic schema matching with Cupid. In: Proc. 27th VLDB Conference. (2001) 49–58
10. Doan, A., Madhavan, J., Domingos, P., Halevy, A.: Learning to map ontologies on the Semantic Web. In: Proc. World-Wide Web Conference. (2002) 662–673
11. Melnik, S., Garcia-Molina, H., Rahm, E.: Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In: ICDE. (2002) 117–128
12. van Keulen, M., de Keijzer, A., Alink, W.: A probabilistic XML approach to data integration. In: ICDE. (2005)
13. Gal, A., Anaby-Tavor, A., Trombetta, A., Montesi, D.: A framework for modeling and evaluating automatic semantic reconciliation. VLDB Journal **14** (2005) 50–67
14. Rizopoulos, N.: Automatic discovery of semantic relationships between schema elements. In: ICEIS (1). (2004) 3–8
15. McBrien, P., Poulovassilis, A.: Data integration by bi-directional schema transformation rules. In: Proc. ICDE, IEEE (2003) 227–238
16. Shafer, G.: A mathematical theory of evidence. Princeton University Press (1976)