Reconciling Agent Models

Jose-Luis Ambite and Craig A. Knoblock
Information Sciences Institute and Computer Science Department
University of Southern California
4676 Admiralty Way,
Marina del Rey, CA 90292, U.S.A.
{jambitem, knoblock}@isi.edu

1 Introduction

Current computer networks offer the possibility of accessing and sharing widely distributed information. The SIMS architecture [2] offers a system for sharing information between multiple, heterogeneous, and distributed information agents. A SIMS information agent is given a model of some domain, models of the contents of information sources (IS) that refer to such domain, and a description of how IS concepts relate to those of the domain model. Then, it uses these models to find the most appropriate set of information sources that satisfy a given query. These models are expressed in a rich knowledge representation system (Loom [3], a description logic).

This paper addresses the problem of automatically reconciling the model of an agent with that of its information sources (other agents, data or knowledge bases, etc), and adapt these models to the evolution of the IS's. The main idea is to analyze the actual extensions (i.e., the current instances) of the IS's, checking the consistency of this data with the definitions in the domain and IS models. This processing will generate novel concepts that will prove useful in improving the accuracy of these models and the efficiency of the system. A prototype for model reconciliation has been implemented using a SIMS mediator that accesses relational databases.

The outline of this extended abstract is as follows. First, we present the different types of new concepts that may appear when integrating IS's. Then, we describe how to generate (learn) useful descriptions of these concepts. Third, we show how such concepts can be used to refine the models, adapt to changes in the contents of the IS's and improve the efficiency of query answering. Finally, we discuss future work.

2 Introducing Novel Concepts in the Domain Model

This section describes the types of interesting concepts that may arise as a consequence of the model reconciliation process, and how they are related to the pre-existing concept lattice (domain and IS models).

2.1 Refining the Model of a Single Information Source

The first verification phase consists in checking that the extension of an information source concept (ISC) actually satisfies the definition of the corresponding domain model concept (DMC). A domain concept can be more complex than the ISC in several ways:

- more roles can be defined for the DMC that do not appear in that particular ISC (their role-fillers may come from other ISC's, or may be computed from those values).
- the ranges of the DMC's roles can be more strictly defined than those of the ISC. For example, an attribute age in the IS model could be defined as just a number, while in the domain model there may exist a more specialized concept of age, as a non-negative integer less than 130.
- there might be additional inter-role constraints. For example, the clinical history of a patient cannot be greater that his age.

If instances that do not satisfy the definition of the DMC are found in the data coming from ISC, the model will be changed as shown in Fig 1. A new concept (C1) is automatically created to represent whatever the actual contents of the information source are (indicated by the dashed line). C1 subsumes (indicated by the solid arrows) both the original DMC and the newly discovered concept (NotDMC). Moreover, DMC and NotDMC form an exhaustive partition of C1 (indicated by the thin curved double line).

As a concrete example, assume ISC1 represents information about cars stored in some database, and it has been related with a concept of *car* (DMC1) in the domain model, in which definition, among other typical constraints, it is stated that the maximum number of passengers (*max-num-pax*) is 5. However, after analyzing ISC1's actual extension, a set of instances, characterized by the constraint: $8 \leq$ *max-num-pax* ≤ 12, does not satisfy the definition of DMC1. This set might actually be *minivans* (which presence in ISC1 might have been unknown to the model designer). Therefore, C1 would represent *passenger vehicles*, NotDMC1 would be *minivans*, and DMC1 the original notion of *car*. Alternatively, a small cardinality of the concept NotDMC1 may signal errors in the database.

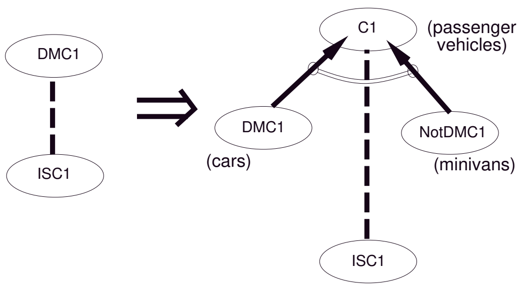

Figure 1: Model Refinement when data is inconsistent with initial definition

2.2 Reconciling Two Related Information Sources

Assume that we have two information source concepts (ISC1, ISC2) mapped to a domain model concept DMC3. An example, used throughout the paper, is shown in Figure 2. The dashed lines represent the semantic equivalence between concepts (and also among their roles, which was omitted for clarity). More generally:

- ISC1 is a concept coming from information source IS1. Assume it is a conjunction of roles (K1, A1, ...Am) where K1 is the key. In the example in Figure 2, the concept *patients* from DB1 has the roles *patients.name* (the key), *patients.age* and *patients.doctor*.

- ISC2 is a concept coming from information source IS2, with roles (K2, B1, ...Bm). In the example in Figure 2, the concept *pats* from DB2 has the roles *pats.name* (the key), *pats.age* and *pats.doctor-name*.

- DMC3 is a domain model concept (K3, R1, ..., Rm, ...Rn), where K3 is the key role, and R1, ..., Rn, other roles ($m \leq n$). In the example in Figure 2, the concept *patient* in the DM has the roles *name* (the key, inherited from the concept *person*), *age* (also inherited), *patient-of*, *diagnosis* and *doctor-name*.

- Equivalences: $K1 =_{ext} K3 =_{ext} K2$. That is, both ISC1 and ISC2 provide the complete extension of object instances for the concept DMC3. This definition of extensional equivalence ($=_{ext}$) is useful when we want to be certain that we retrieve *all* the data that satisfies a given query.

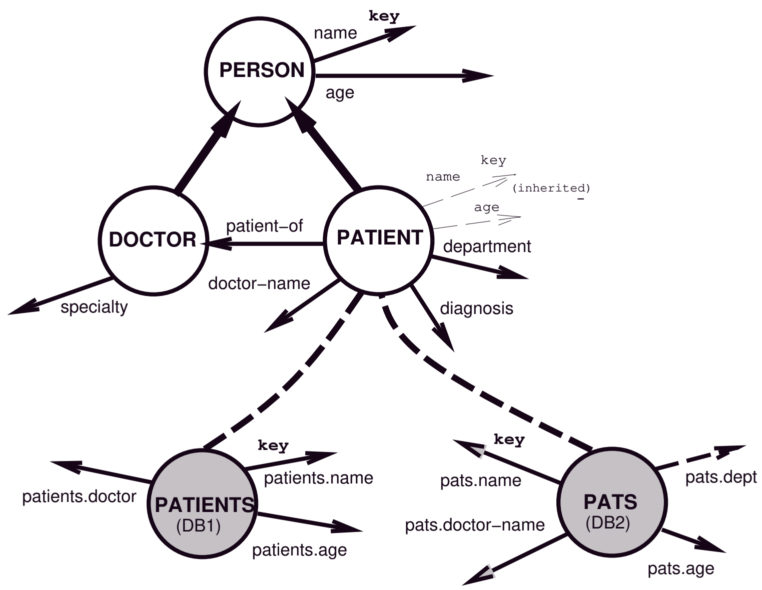

Figure 2: Modeling Example

2.2.1 Refinement Algorithm

Assume that the concepts have satisfied the verification of the 1-to-1 mappings described in Section 2.1. Now, the system compares the actual extensions of the ISC's in order to verify this initial model or propose its modification.

1. Obtain ranges of key roles of the ISC's, which will allow the system to uniquely identify the objects instances involved.

2. Compare key sets K1 (of ISC1) and K2 (of ISC2). There are 4 cases:

   a. $K1 \subset K2$

   K1 is contained in K2. Then, the model changes as shown in Fig 3:

- A concept C1 is created in the domain model representing the contents of ISC1, including in its definition a description that explains why it is a specialization of DMC3 (see Section 3).
- State that DMC3 subsumes C1, delete equivalences among ISC1 and DMC3, and restate them among ISC1 and C1.
- ISC2 equivalences remain unaltered.

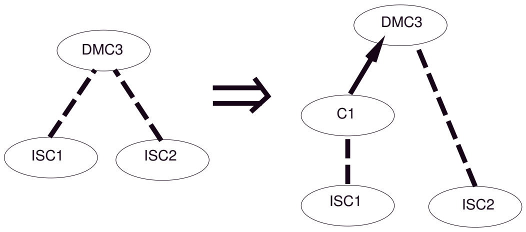

Figure 3: Domain Model Refinement when $K1 \subset K2$

b. $K1 = K2$

The extension of both key sets is identical. Then, we should proceed to check for each of the remaining attributes of the IS's whether their corresponding values agree. If they do, the initial integration remains valid. Otherwise, the conflicting tuples should be separated and,recursively, new concepts formed and explained (see Section 3).

c. Overlapping key sets

If the key sets are overlapping, i.e., $(K1 \cap K2 \neq \emptyset) \wedge (K1 \not\subseteq K2) \wedge (K2 \not\subseteq K1)$. Then, the following novel concepts are formed:

- C1-C2: set difference of key sets K1 and K2.
- C2-C1: set difference of key sets K2 and K1.
- C1&C2: intersection of key sets K1 and K2.
- C1 that represents the contents of ISC1.
- C2 that represents the contents of ISC2.

The domain model is modified as in Fig 4, where the thin single curved line means that the subconcepts form a covering of their superconcept (but not a partition). For each of these concepts (as represented by their current extensions) we search for a plausible explanation (see Section 3).

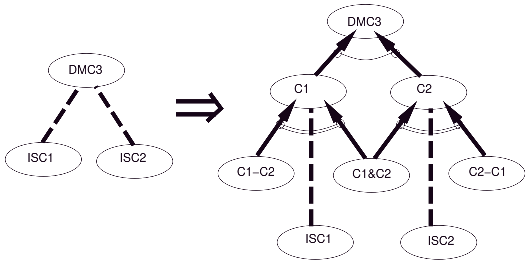

Figure 4: Domain Model Refinement when the key sets are overlapping

d. Disjoint key sets

Although we initially considered the IS's as identical sources for the concept DMC3, they, in fact, have none of their instances in common (however recall that they satisfy the 1-1 mappings of Section 2.1). In this case the model is modified to reflect that the sources form an exhaustive partition[1] for DMC3, as shown in Figure 5.

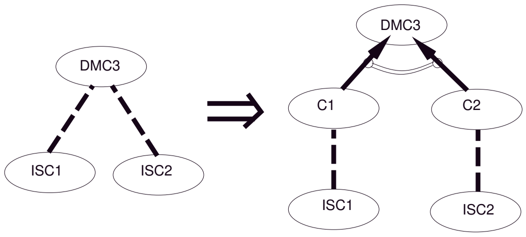

Figure 5: Domain Model Refinement when the key sets are disjoint

2.3 Reconciling Multiple Related Information Sources

The number of interesting sets that could be considered grows rapidly with the number of identical information sources stated. For example, with 3 ISC's, we would have to consider each initial concept, their set differences, pairwise intersections, etc. Nevertheless, this problem is ameliorated in practice due to two factors. First, it is improbable that a great number of IS's could be found in the real world as sources for the same concept (that is, identical, recall the semantics of the equivalences). In second place, not all of this combinatorial number of sets would be found interesting, that is, their descriptions might be too cumbersome to be useful. Thus, only some of them would be represented and reasoned about in the model.

3 Finding Descriptions of the new Concepts

In the previous section, a way of generating novel concepts was outlined. Now, it is necessary to concisely describe these new groupings of instances so that the system can justify declaratively the concept relationships stated in the figures of Section 2. If these explanations are found interesting, they provide a way of naturally specializing the concepts by adding them as constraints in their definitions. The algorithms for discovering these regularities within sets of instances can be some of those found in the literature of database mining/machine learning, suitably modified to take advantage of the richer structure of our domain model. A prototype has been implemented that uses ID3[4] as its learning algorithm.

Consider the integrated model of Figure 2, these are some simple examples of the types of concept descriptions that could be learned from the databases, corresponding to the cases outlined in Subsection 2.2.1:

a. $K1 \subset K2$:

It might be the case that DB1 is the database of just a department in a hospital, for example, Pediatrics, while DB2 contains patients from the whole hospital. An interesting case occurs

[1] Note that all the instances of concept DMC3 come from either ISC1 or ISC2

when some information not present in the "subset" database (DB1) is present in the "superset" database (DB2). Assume that the concept *patient* in the domain model had a role *department* (coming from *pats.dept* in DB2) that shows to which department each patient belongs. Then, we could use this extra role to build an explanation, by noting that all tuples in DB1 have the same value *in DB2*: *pats.dept* = '*Pediatrics*'. Our prototype currently provides a description expressed in terms of the domain model over the *common* roles of the concepts involved (i.e., the roles *name*, *age* and *doctor-name*, for the example in Figure 2). The induced description is:

(defconcept Infant-Patient :is-primitive (:and Patient (< Age 10.0)))

c. Overlapping key sets:

Assume a situation in which DB1 and DB2 are supposed to have all the general medicine patients in the hospital, but the actual extension of DB1 includes also pediatrics patients, while geriatrics patients are in DB2. Analogously to the previous case, the instances in C1-C2 could be described by $age \leq 10$ (years old), or *doctor.specialty* = '*Pediatrics*', and those in C2-C1 by $age \geq 60$ or *doctor.specialty* = '*Geriatrics*'.

d. Disjoint key sets:

This case could result if DB1 was actually the Geriatrics department database and DB2 Pediatrics'. Note that the system would need to look for descriptions as in case c., because the extra information of an attribute like *pats.dept* of case a. is unlikely to be available.

4 Benefits of the Refinement

The advantages of introducing these novel concepts as discussed in the previous sections are:

- Increased accuracy of the knowledge represented in the system:

  First, these concepts provide a more precise picture of the current information available to the system. Second, this mechanism adapts automatically to the evolution of the information sources, which contents may semantically drift from the original domain model mappings. Finally, human designers may revise these concepts to both refine the domain model and detect errors in the databases.

- Increased efficiency of query processing:

  Extensions to the query reformulation and query access planning mechanisms of SIMS will look for those concepts that preserving the semantics of the user query (that is, retrieving the same instances as the original query), and being grounded in available information sources, eventually yield a cheaper query plan. These new concepts provide better options for retrieving the desired data.

Some examples, from the domain in Figure 2, that explore these benefits are:

a. Contained

Assume the query asks for those patients treated by doctors that are pediatricians. That query would involve a join between the information source that keeps the specialties of the doctors and the information in one of the patients IS's. If the system chooses *pats* in DB2 a greater number of tuples is transmitted through the network and have to be tested in the local join. On the other hand, if the query is reformulated using the new concept (C1), it will be found to be better answered by DB1, thus saving in communication and processing cost.

b. Equivalent

The IS providing cheaper access would be selected. For example, DB1 might be chosen because it is installed in a more powerful processor, or being linked with a faster network.

c. Overlapping

Assume the user query asks for those infant patients suffering from measles. The initial model will miss completely the answer if it chooses DB2, that contains no infant patients. But with the refined domain model the query would be accurately directed to DB1. If the system was just aware of the overlap of both ISC's and decided to play safe by querying both databases, it would waste resources communicating with DB2 that actually has no information relevant to the query. Thus, we also gain in performance.

d. Disjoint

Assume any of the two previous queries is asked by the user. Again, only one of the IS's needs to be accessed (DB1), communicating with DB2 would be wasteful.

5 Discussion

We have presented a framework for reconciling agent models. We have introduced a set of novel concepts that refine these models, suggested ways to generate useful descriptions of them, and explained their utility to improve the accuracy and efficiency of a system of information agents. An prototype system has been implemented.

Future work will include exploring which learning/knowledge discovery algorithms provide more useful concept descriptions (for example, results from [1] may be promising) and how to perform the extensional analysis incrementally to better keep track of the evolving information sources.

References

[1] William W. Cohen and Haym Hirsh. Learning the classic description logic: Theoretical and experimental results. In Erik Sandewall Jon Doyle and Pietro Torasso, editors, *Priciples of Knowledege Representation And Reasoning. Proceedings of the Fourth International Conference*, pages 121–133, Bonn, Germany, 1994.

[2] Craig Knoblock, Yigal Arens, and Chun-Nan Hsu. Cooperating agents for information retrieval. In *Proceedings of the Second International Conference on Cooperative Information Systems*, Toronto, Canada, 1994.

[3] Robert MacGregor. The evolving technology of classification-based knowledge representation systems. In John Sowa, editor, *Principles of Semantic Networks: Explorations in the Representation of Knowledge*. Morgan Kaufmann, 1990.

[4] J.R. Quinlan. *Inductive inference as a tool for the construction of efficient classification programs*. Tioga, Palo Alto, CA, 1983.