MEDINFO 2004
M. Fieschi et al. (Eds)
Amsterdam: IOS Press

Challenges in Precisely Aligning Models of Human Anatomy Using Generic Schema Matching

Peter Mork[abc], Rachel Pottinger[ab], Philip A. Bernstein[ab]

[a]*Microsoft Research, Redmond, WA, USA*
[b]*Computer Science & Engineering, University of Washington, Seattle, WA, USA*
[c]*Biomedical & Health Informatics, University of Washington, Seattle, WA, USA*

Abstract

This paper describes how we used generic schema matching algorithms to align the Foundational Model of Anatomy (FMA) and the GALEN Common Reference Model (CRM), two large models of human anatomy. We summarize the generic schema matching algorithms we used to identify correspondences. We present sample results that highlight the similarities and differences between the FMA and the CRM. We also identify uses of aggregation, transitivity, and reification, for which generic schema matching fails to produce an accurate mapping and present manually constructed solutions for them.

Keywords:

Information Management, Databases, Unified Medical Language System, Ontology Alignment

Introduction

Anatomy is the foundation of both modern medicine and biological research. The extensive nomenclature developed over the centuries allows physicians to share observations and diagnoses. Human anatomy has been encoded in two of the world's largest symbolic models: the Foundational Model of Anatomy (FMA) and the GALEN Common Reference Model (CRM). The goal of our project was to compare and contrast these models. This paper reports on the challenges encountered in producing an alignment.

The FMA [1] is being developed at the University of Washington under the guidance of Dr. Cornelius Rosse. The goal of this project is to capture all of human anatomy in precise detail. The intent is to support a wide variety of applications, including image retrieval and rendering, teaching, and natural language querying. The copy of the FMA we received consists of roughly 59,000 concepts organized in four main hierarchies: parts, subclasses, branches and tributaries. The model also contains over 100 types of relationships; roughly 1.6 million relationships interconnect the core concepts.

The GALEN CRM [2] was developed at the University of Manchester by Dr. Alan Rector, et al. GALEN represents a collection of technologies and resources designed to facilitate knowledge reuse across clinical applications. Given this smaller focus, the CRM consists of only about 24,000 concepts expressed in a description logic [3] (a framework for automatically organizing an inheritance hierarchy). Similar to the FMA, the CRM includes an array of hierarchies: parts, subclasses and branches (including tributaries). Unlike the FMA, the CRM framework includes relationship specialization. For example, the part hierarchy in the CRM is represented by several specializations of HasDivision, which corresponds to generic part in the FMA. The CRM contains roughly 913,000 relationships among the core concepts.

We present three main contributions. First, we relate model alignment to generic schema matching. Second, we describe how to adapt existing generic schema matching algorithms to identify correspondences between the FMA and the CRM and highlight some of the similarities and differences (between the models) identified using this approach. Third, we identify three common scenarios where precise mappings were not found using generic schema matching: aggregation, transitivity and reification. For each of these we provide manually constructed mappings that are richer than simple correspondences. These richer mappings allow precise expression of the relationships between the schemas.

These mappings are interesting in and of themselves; they establish points of consensus. Moreover, a precise mapping is required for more advanced operations like identifying differences or model merging.

The remainder of the paper is organized as follows. In the next section we present an overview of our meta-model and describe how we imported the FMA and the CRM into this framework. We next sketch the matching algorithms we used and then present sample match results. Finally, we describe the challenging scenarios and the solutions we identified.

Background

For the purposes of this paper, a model is the description of some system, theory or phenomenon. It is a tool for simplifying reality and encoding it in a symbolic and systematic way. For example, a schema in a database management system is a model that describes the data to be stored. In knowledge engineering, a model is a description of an aspect of the world, to be queried and manipulated directly.

A model is expressed in a meta-model, which describes the space of valid models. The FMA is written in Protégé-2000 [4], a frame-based meta-model; the CRM is written using a description logic [3] meta-model. To match two schemas, they need to be expressed in the same meta-model. We therefore imported the FMA and the CRM into a common meta-model called Vanilla, which is based on prior work [5].

Vanilla represents a model as a graph. Its nodes correspond to elements, which include classes and instances. Its edges represent relationships between elements. It has eight kinds of relationships: Is-a (class generalization), Type-of (instantiation), Contains (sub-structure), Can-contain (template structure), Has (reference), Can-have (template reference), Related-to and Maps-to. Figure 1 displays the subset of Vanilla edges used in this paper and shows a simple relational database encoding. The Hospital DB is an instance of the class Databases (sans serif indicates graph elements). It contains two tables (Patients and Doctors), which in turn contain columns. The column Sees is related to the column SSN.

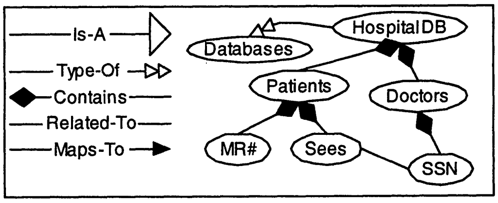

Figure 1 - Vanilla relationships and a sample model

We now describe how to encode the FMA and the CRM in this meta-model: The basic construct in the FMA is a (frame, slot, value) triple. For example, the FMA asserts (Heart, generic part, Cardiac valve), which indicates that the Cardiac valve is a generic part of the Heart. Since the meta-model has a non-extensible collection of relationship types, this assertion cannot be encoded directly. Instead, it is reified as shown in figure 2: The relationship becomes an explicit element contained in the Heart element. This element is of type generic part and references Cardiac valve. One could read this as, "The heart contains a generic part relationship whose value is the cardiac valve." CRM constructs are dealt with similarly.

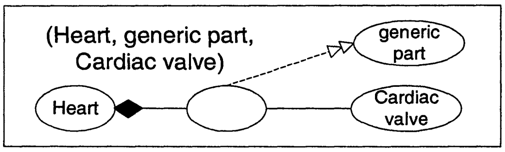

Figure 2 - Sample encoding of a Protégé-2000 triple in Vanilla

A Vanilla *mapping* is a first-class model, which can consist of all Vanilla elements and relationships. In addition, every element in a mapping between models A and B can participate in one or more Maps-to relationships with elements from A or B. Expressing mappings as first-class objects allows more precise correspondence description. For example, one can indicate that the Full Name element in one model contains the First Name and Last Name elements from another model. When we describe manually constructed mappings that address the challenges identified (below), we show how the additional detail provided by first-class mappings can be leveraged when relating models that differ in their level of detail.

Methods

Our approach for finding similarities and differences between the FMA and the CRM can be summarized as follows: 1) Import both models into a common meta-model. 2) Produce a mapping using generic schema matching techniques. 3) Identify differences by comparing the original models to the mapping.

We described in the previous section how step one was accomplished. Due to space limitations, we will only describe step two here. Details of the "diff" operator used in step three can be found in prior work [6].

Given two input models, a schema matching algorithm returns pairs of elements that are similar. The similarity of two elements can be based on various criteria including: name, data type, neighborhood, data instances, etc. Many algorithms have been proposed, a survey [7] of which appears elsewhere.

Given the size of the FMA and the CRM, we needed an efficient algorithm. We modified a variant of Cupid [8], which exploits name and structure similarities, by adding a hierarchical matcher from LSD [9]. Our algorithm has 3 phases: lexical matching, structure matching and hierarchical matching. The final output is a set of 1:1 correspondences between elements. This approach is similar to the one presented by Zhang and Bodenreider [10], which also establishes lexical and structural matches using a rather different non-generic algorithm. Zhang and Bodenreider also augment their system with inferred knowledge.

The first phase converts each concept name to a collection of terms. Using the SPECIALIST Lexicon [11] published by the National Library of Medicine, a string is converted to a collection of normalized terms drawn from a thesaurus. These tools account for differences in punctuation, spelling, and conjugation, as well as handling synonyms. Cupid takes these collections of terms and computes a similarity score based on term overlap and usage information.

The second phase uses a variant of similarity flooding [12], where the similarity of a given pair of elements is influenced by the similarity of its neighbors. The intuition is that if two concepts (or relationships) are *used* similarly, then they probably *are* similar. In practice, this phase aligned relationships quite well (e.g., detecting that HasComponent is related to generic part). Briefly, it assigns to each pair of reified elements a similarity score equal to the average of its neighbors. This score is *back propagated* to the neighbors, allowing their similarity scores to slowly increase. Given sufficient evidence, the similarity of two elements approaches one (a perfect match).

Finally, the third phase uses the inheritance hierarchies to align super-classes. The similarity between two (non-leaf) classes is iteratively set to the average similarity exhibited by children, grandchildren and great-grandchildren. Intuitively, two super-classes are the same if they have matching descendants.

Sample Results

In this section we present some statistics on our matches. To highlight similarities and differences between the models, we also present two good matches and one bad match. The quality of a match is measured by the number of (frame, slot, value) assertions agreed on by both the FMA and the CRM.

As a baseline, a simple string comparison identifies merely 306 matches (out of a possible 24,000 based on the smaller model). Because the CRM uses terms without spaces (CaMeL case), by adding relevant spaces to CRM terms and ignoring case, 1834 matches can be found using string comparison.

Using the lexical tools described above increases the number of matches to 3503. Structure matching adds 64 matches. At first, this seems disappointing, but prior to structure matching, almost no relationship elements were matched. For example, tributary (from the FMA) did not match HasBranch (from the CRM) and generic part did not match HasDivision. One notable exception is that contains matched contains. With these 64 additional matches, we matched 875 reified relationships (anonymous elements like in figure 2). Finally, hierarchical matching, not used by Zhang and Bodenreider, increased the total number of element matches to 3780. This is comparable to the 2353 matches found by Zhang and Bodenreider [10].

Sample results are in figures 3 and 4. Figure 3 shows terms and relationships that are in both models and hence the match is the identity. This holds for both the elements and the relationships between them. For example, both models assert that one of the branches of the Abdominal aorta is the ovarian artery.

Abdominal aorta branches:
{ovarian artery, inferior phrenic artery, inferior mesenteric artery, superior mesenteric artery, renal artery, common iliac artery, lumbar artery, median sacral artery}

Median nerve innervates:
{flexor digitorum superficialis, flexor carpi radialis, palmaris longus, pronator teres, flexor digitorum profundus}

Figure 3 - Sample good matches; all elements and relationships are present in both the FMA and the CRM

Figure 4 shows sample assertions from the FMA and from the CRM concerning the Lung. Both models include parts of the Lung, but they disagree on every part. Some of the discrepancy stems from an inaccurate mapping: lobe of lung should match lobe. Others are due to different modeling decisions: The models do not agree on the relationship between the lung and parenchyma, nor on the Arterial supply unless the bronchial arterial trunk should match the bronchial artery. Moreover, the FMA asserts that the lung is Contained in the thoracic cavity; the CRM claims the lung Is contained in the pleural membrane.

Based on our observations, the best matches tend to agree on supply relationships (arteries, nerves and veins). This may be caused by general agreement of the (directly observable) relationships between anatomical structures and arteries, nerves and veins. Generic schema matching algorithms succeed at identifying these matches for two reasons: 1) Anatomists agree, for the most part, on the names of the structures involved, and 2) the models express this information at the same level of detail (i.e., a single relationship). There is less agreement on partitive relationships, which the modelers can express at differing levels of detail (see the next section). In addition, the modelers may partition according to different criteria.

FMA: Lung	**CRM: Lung**
General part:	**Has division**:
{visceral pleura, lung	{apex, base of structure,
parenchyma, lower	lobe, part of pleura,
lobe of lung, lobe of	pulmonary artery, hilum,
lung, apex of lung	pulmonary vein, ...}
(viewed anatomically),	**Is made of**:
...}	parenchyma
Arterial supply:	**Is served by**:
bronchial arterial trunk	{bronchial artery,
Venous drainage:	pulmonary artery,
bronchial venous tree	pulmonary vein}
Bounded by:	**Bounds space**:
surface of lung	mediastinum
Contained in:	**Is contained in**:
thoracic cavity	pleural membrane

Figure 4: Sample bad match; the models do not agree on the values for any of the relationships.

Figure 4 - Sample bad match; the models do not agree on the values for any of the relationships

Challenges

We now present several common cases in which generic schema matching failed to produce a desirable result. A common feature of these cases is that the two models express different granularity of detail. As a result, the relationships between the models are not satisfactorily captured by simple correspondences between elements generated by generic schema matching algorithms. This is an advantage of our approach over the Zhang and Bodenreider work [10]; they do not consider first class mappings and hence do not resolve these types of conflicts. We illustrate these scenarios with specific examples, but the techniques presented are more generally applicable.

Aggregation: A common difference between models is the granularity with which classes and relationship types are expressed. The CRM includes a large hierarchy of relationships. For example, the many children of HasDivision (e.g., HasLayer), correspond to a single relationship (generic part) in the FMA.

The FMA can also be more precise than the CRM, as shown in figure 5. In this example, four relationships in the FMA correspond to a single relationship in the CRM. The FMA does not use a relationship hierarchy, so there is no parent relationship that aggregates the four supply relationships. A naïve mapping (not shown) would relate the single relationship in the CRM to each of the four FMA relationships.

Figure 5 shows a better mapping. Rather than just correspondences, the mapping explicitly states that the supplied by relationship in the CRM corresponds to the super-class of the FMA relationships. Note that this relationship is expressed only in the mapping—neither the FMA nor the CRM are modified.

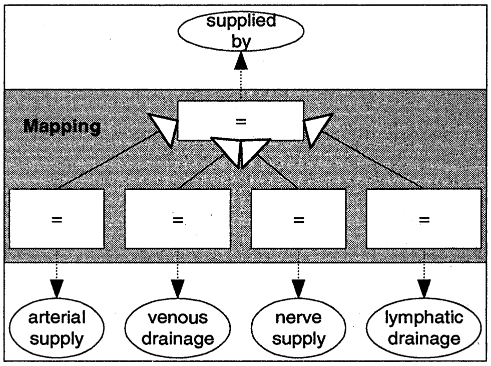

Figure 5 - Hand-constructed mapping that relates supplied by (from the CRM) to four similar relationships (from the FMA)

Transitive Relationships: Many of the relationships in these models are transitive, such as parts, branches, and tributaries. The ramification of transitive relationships is that the two models may agree that some element is part of another, but they may express this knowledge via differing degrees of indirection.

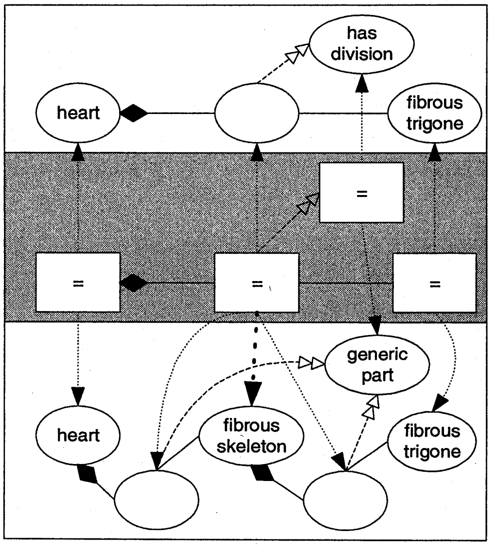

Figure 6 - Using a single mapping element to align transitive relationships; this mapping erroneously relates the fibrous skeleton to a relationship

Consider the example in figure 6, which displays a naïve mapping. In both models, the fibrous trigone is part of the heart. However, in the FMA, the fibrous trigone is only indirectly part of the heart; more specifically, the fibrous trigone is part of the fibrous skeleton (of the heart), which is part of the heart.

While more precise than merely equating the heart and fibrous trigone, the mapping in figure 6 has the unfortunate effect of stating that the fibrous skeleton corresponds to a relationship between the heart and the fibrous trigone. Deleting the edge between the fibrous skeleton and the mapping improves the mapping, but it is still not clear how the CRM relationship corresponds to the two relationships in the FMA.

The mapping in figure 7 demonstrates more precisely the correspondences between the two models: fibrous skeleton does not have a correspondence in the CRM. In addition, the model indicates that the relationship in the CRM can be broken into two sub-relationships.

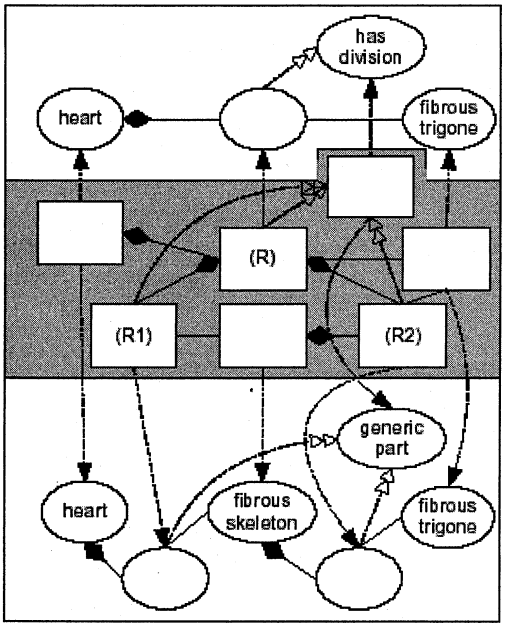

Figure 7 - A more precise mapping for transitive relationships; relationship (R) contains two sub-relationships (R1 and R2).

Reified Relationships: The final example, shown in figure 8, arises from the use of reified relationships in the FMA to encode additional information about a given relationship. For example, wall of heart is part of heart, but moreover wall of heart is an unshared part (of the heart). Because Protégé-2000 does not allow attributes to be added to relationships, the relationships must be reified as first-class instances.

The CRM does not provide this level of detail. As shown in figure 8, when these relationships are imported into Vanilla, the number of intermediate elements between heart and wall of heart differ. The correct mapping is not conceptually difficult; each model contains a single relationship relating heart and wall of heart. This mapping is not automatically identified because structural matching algorithms require that the neighboring elements (shown in bold) participate in the mapping, which is not true with reified relationships. Zhang and Bodenreider [10] avoid this difficulty by manually removing reification.

Conclusion

We have demonstrated that generic schema matching algorithms can match two large models of anatomy, the FMA and the CRM. We have shown the strengths of these algorithms, and their limitations, especially when the models express knowledge at differing levels of detail (as one would expect when the sizes of the models differ substantially). We have provided hand-constructed mappings for three common situations in which generic schema matching algorithms do not provide precise mappings because the models being matched differ in their levels of detail: aggregation, transitivity and reification..

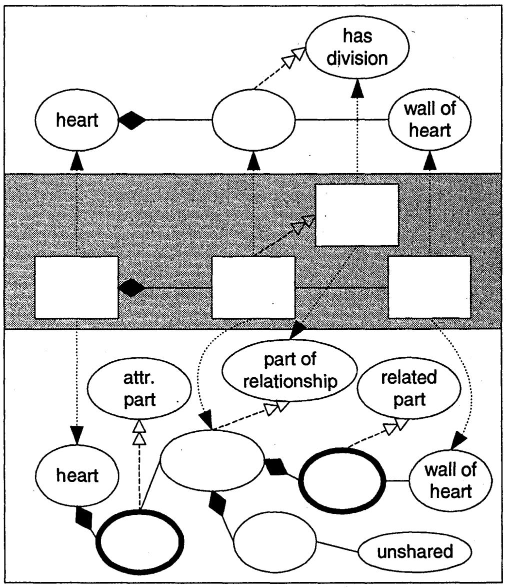

Figure 8 - Sample mapping for a reified relationship

Looking forward, we can begin to design algorithms more capable of identifying the precise relationships between two models. This will be helped, in part, by having anatomists further evaluate the quality of the simple correspondences we have identified so far.

Acknowledgements

We would like to thank Kurt Rickard and Jeremy Rogers for their help in obtaining copies of the FMA and the CRM, respectively.

References

[1] Rosse C, Mejino JL, Modayur BR, Jakobovits R, Hinshaw KP, Brinkley JF. Motivation and organizational principles for anatomical representation: The Digital Anatomist symbolic knowledge base. *JAMIA* 1998;5(1):17–40.

[2] Rector AL, Gangemi A, Galeazzi E, Glowinski AJ, Rossi-Mori A. The GALEN core model schemata for anatomy: Towards a re-usable application-independent model of medical concepts. In: *MIE; 1994;* Lisbon, Portugal; 1994. p. 229–233.

[3] Zanstra PE, van der Haring EJ, Flier F, Rogers JE, Solomon WD. Using the GRAIL language for classification management. In: *MIE; 1997*; Thessaloniki, Greece; 1997. p. 897–901.

[4] Musen M, Crubézy M, Fergerson R, Noy NF, Tu S, Vendetti J. Protégé-2000. Stanford Medical Informatics (Stanford, CA). http://protege.stanford.edu/

[5] Pottinger RA, Bernstein PA. Merging models based on given correspondences. In: *VLDB*; 2003 September 9–12; Berlin, Germany: Morgan Kaufmann; 2003.

[6] Bernstein PA. Applying model management to classical meta data problems. In: *CIDR*; 2003 Jan 5–8; Asilomar, CA; 2003.

[7] Rahm E, Bernstein PA. A survey of approaches to automatic schema matching. *VLDBJ* 2001;10(4):334–350.

[8] Madhavan J, Bernstein PA, Rahm E. Generic schema matching using Cupid. In: Apers PMG, Atzeni P, Ceri S, Paraboschi S, Ramamohanarao K, Snodgrass RT, editors. *VLDB 2001*; 2001 Sept 11–14; Roma, Italy: Morgan Kaufmann; 2001. p. 49–58.

[9] Doan A, Domingos P, Halevy AY. Learning to match the schemas of databases: A multistrategy approach. *Machine Learning* 2003;50(3):279–301.

[10]Zhang S, Bodenreider O. Aligning representations of anatomy using lexical and structural methods. In: *AMIA*; 2003 November 8–12; Washington, DC: AMIA; 2003.

[11]SPECIALIST lexicon. National Library of Medicine (NLM) (Bethesda, MD). http://www.nlm.nih.gov/pubs/factsheets/umlslex.html

[12]Melnik S, Garcia-Molina H, Rahm E. Similarity flooding: A versatile graph matching algorithm. In: *ICDE*; 2002; San Jose, CA; 2002.

Address for correspondence

Peter Mork
University of Washington, Computer Science & Engineering
Box 352350
Seattle, WA 98195
USA
pmork@cs.washington.edu