Automatic ontology mapping for agent communication

F. Wiesman
wiesman@cs.unimaas.nl
IKAT, Universiteit Maastricht

N. Roos
roos@cs.unimaas.nl
P.O.Box 616

P. Vogt
p.vogt@cs.unimaas.nl
6200 MD Maastricht The Netherlands

ABSTRACT

Agent communication languages such as ACL and KQML provide a standard for agent communication. For the protocol and the language used in the communication, several standards are available. This is not the case for the ontology used in the communication. The ontology depends on the subject of the communication. Since the number of subjects is almost infinite and since the concepts used for a subject can be described by different ontologies, the development of generally accepted standards will take a long time. This lack of standardization, which hampers communication and collaboration between agents, is known as the *interoperability* problem. To overcome the interoperability problem, an approach that enables agents to learn a mapping between their ontologies will be proposed.

Keywords

Ontologies and knowledge management, Agent communication languages and protocols, Cooperation

1. INTEROPERABILITY

An ontology is used to describe the meaning of concepts in agent communication. This ontology depends on the subject of the communication. Since the number of possible subjects is almost infinite and since the concepts used for a subject can be described by different ontologies, the development of generally accepted standards will take a long time. This lack of standardization, which hampers communication and collaboration between agents, is known as the *interoperability* problem [4, 7].

In order to reach interoperability, two problems must be dealt with, namely: *structural heterogeneity* and *semantic heterogeneity* [7]. Structural heterogeneity concerns the different representations of information. Information described by the same ontology can be represented in different ways. This is a problem for heterogeneous databases but not for agents. In a multi-agent system an ontology is the basis for communication. The actual way information is stored by an agent is shielded from the environment by the agent.

Semantic heterogeneity concerns the intended meaning of described information. Information about, for instance, persons can be described by different ontologies. We distinguish the following differences between ontologies: (1) different semantic structures, *structural conflict*, (2) different names for the same type of information or the same name for (slightly) different types of information, *naming conflicts*, and (3) different representations of the same data, *data conflicts*. The data conflict can be refined in conflicts because of *different units*, conflicts because of *different precision*, and conflicts because of *different expressions* (e.g., using 'van den Herik' or 'Herik, van den' to describe a person's family name).

Figure 1 illustrate some forms of semantic heterogeneity. In ontology 1, 'street' also describes the house number and 'phone number' describes the country code, the area code, and the local number. In ontology 2, 'phone number' only describes the local number. The 'area code' and the 'country code' are stored with, respectively, the city and the country. Since the two ontologies have different structures, we have a structural conflict.

Naming conflicts between the two ontologies present a more severe problem. Different concept names are used for the same type of data (e.g., 'first name' and 'christian name'). Moreover, the same concept name is used for slightly different types of data; e.g., 'street'. In ontology 1 'street' denotes both the *street name* and the *house number* while in ontology 2 it only denotes the *street name*. Hence, in order to reach interoperability, we must be able to split and merge data fields. For instance, the concept 'street' in ontology 1 containing the data 'Castle Lane 1' must be split into 'Castle Lane' and '1' in order to map 'street' in ontology 1 to 'street' and 'number' in ontology 2. The inverse mapping requires merging 'Castle Lane' and '1'.

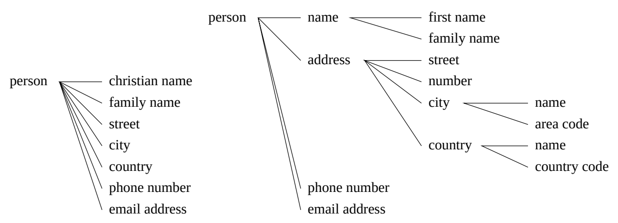

Figure 1: Ontologies 1 and 2.

We can conclude that to reach interoperability we have to find a mapping from the concepts of one ontology to the concepts of another ontology while instances of the concepts can be split and merged. None of the approaches proposed in the literature, e.g. [1, 2, 4, 3, 6], offer an adequate solution.

2. LEARNING ONTOLOGY MAPPINGS

Suppose that agent 1 wishes to know the phone number and email address of some persons. Agent 1 knows that the information is (probably) available in a database managed by agent 2. Therefore, agent 1 contacts agent 2. In order for agent 1 to put forward its request, the agents first have to establish whether both use the same ontology or whether they use an ontology of which the other agent knows how to map it on its ontology. If the agents use different ontologies and if no mapping is known, the agents should try to establish a mapping. The way the agents establish a mapping is inspired by language games [5].

To illustrate the idea behind using language games for ontology mapping, suppose that two agents wish to communicate about a concept such as a 'person'. Moreover, the agents use different conceptualizations of the concept 'person' (as depicted in Figure 1), and some persons are known by both agents.

A concept such as a 'person' may consists of a hierarchy of sub-concepts. For the *primitive* concepts in this hierarchy, an instance specifies the actual data values. For example, an instance could be a person called 'Haddock', who lives at 'Castle Lane 1, Marlinspike', with phone number '421'. By finding an instance of the concept 'person' known by both agents, the agents determine *joint attention*. This joint attention will be the basis of the language game.

To establish the joint attention, agent 1 produces an utterance containing a unique representation of a concept and instance of the concept. Agent 2, upon receiving the utterance, investigates whether it has a concept of which an instance matches to a certain degree the communicated instance. To do so, agent 2 measures the proportion of words that two instances have in common. The instance with the highest proportion of corresponding words, forms, together with the communicated instance, the joint attention – provided that the correspondence is high enough.

After establishing joint attention, agents 2 tries to establish a mapping between the primitive concepts that make up the concept. To do so, agent 2 needs an utterance from agent 1 and itself. Each of these utterances uniquely represents the primitive concepts of the concept followed by corresponding the data of an instance of the concept. The used representations of the primitive concepts can be any symbol. The only thing that is required is that each representation uniquely represents a primitive concept. Hence, the structure of the ontology plays no role.

Next, agent 2 tries to establish associations between the different primitive concepts. Agent 2 generates associations between the primitive concepts of the two utterances on the basis of the proportion of corresponding words in pairs of primitive concepts, one from each utterance. Possible associations are:

field $x \leftarrow$ field y.
field $x \leftarrow$ field y, split(s), first.
field $x \leftarrow$ field y, split(s), last.
field $x \leftarrow$ field y, field z, merge (t).

Here, the operator field denotes the selection of a primitive concept where x, y, and z represent the primitive concepts to be selected. As explained in Section 1, the operators split divides a data field into two sub-fields using the separator s to determine the point of division. We consider the following separators: ' ', ',', ';', and TC (a type change, i.e., a change from letters to digits or vice versa). After splitting a data field the operators first and last can be used to select either the first or the last sub-field. The operator merge takes two data fields and merges them into one data field adding the separator t in between. As separators can be added: '', ' ', ',' and ';'. The following illustrates a mapping from agent 1 to agent 2.

field person.has.address.has.number $\leftarrow$ field person.has.street, split(TC), last.

Agent 2 searches through a space of possible associations guided by the proportion of words that instances of concepts have in common. Each new utterance from agent 1 enables agent 2 to update the strength of the associations. After having received a number of utterances, agent 2 may accept certain associations as being correct. Agent 2 has established a complete mapping from agent 1 to itself when it has a unique association for each primitive concept in its ontology.

3. EXPERIMENTS

We have conducted experiments for *a proof of concept*. In our experiments, we tried to establish a mapping between three address databases; of our department database, of the ICCA journal, and of the Belgium-Dutch AI Association. The experiments showed that a mapping can be learned between primitive concepts. However, domain-specific knowledge, such as the structure of (Dutch) names and country-dependent address notations, is required for some cases.

4. REFERENCES

[1] J. Hammer and D. McLeod, 'An approach to resolving semantic heterogeneity in a federation of autonomous, heterogeneous database systems', *Journal for Intelligent and Cooperative Information Systems*, **2**(1), 51–83, (1993).

[2] Tova Milo and Sagit Zohar, 'Using schema matching to simplify heterogeneous data translation', in *Proc. 24th Int. Conf. Very Large Data Bases, VLDB*, pp. 122–133, (24–27 1998).

[3] M. P. Papazoglou, N. Russell, and D. Edmond, 'A translation protocol achieving consensus of semantics between cooperating heterogeneous database systems', in *Conference on Cooperative Information Systems*, pp. 78–89, (1996).

[4] H. S. Pinto, 'Some issues on ontology integration', in *IJCAI-99 workshop on Ontologies and Problem-Solving Methods (KRR5)*, (1999).

[5] L. Steels and P. Vogt, 'Grounding adaptive language games in robotic agents', in *Proceedings of the Fourth European Conference on Artificial Life*, eds., C. Husbands and I. Harvey, Cambridge Ma. and London, (1997). MIT Press.

[6] R. M. van Eijk, F. S. de Boer, and W. van der Hoek, 'On dynamically generated translators in agent communication', *International Journal of Intelligent Systems*, **16**, 587–607, (2001).

[7] H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hübner, 'Ontology-based integration of information - a survey of existing approaches', in *IJCAI 2001 Workshop on Ontologies and Information Sharing*, (2001).