Mutual Online Ontology Alignment

Jun Wang   Les Gasser
Graduate School of Library and Information Science
University of Illinois at Urbana-Champaign
501 E. Daniel St., Champaign, IL 61820, USA
{junwang4, gasser}@uiuc.edu

ABSTRACT

One critical foundation for reliable collective coordinated action in multi-agent systems is the ability to exchange representational objects that can be locally interpreted in ways that make collective sense. Coherent local interpretation of shared representations is needed to create global semantic coherence for the distributed actions of individual agents. The process of autonomously establishing a collective semantics and organization of the concepts in an open domain is not well understood. This paper discusses our motivations and approach to this problem. It presents a specific formalization and algorithm that we call *mutual online ontology alignment* with its rationale, and illustrates the performance of the approach with some simulation experiments.

Keywords: ontology evolution, ontology alignment, co-learning

1. INTRODUCTION

> *"We must be systematic, but we should keep our systems open."* (Alfred North Whitehead, Modes of Thought).

In multi-agent systems research and practice, reliable collective coordinated action rests on two foundations:

F1. Distributed reasoning algorithms whose critical property is their ability to manage the control uncertainty inherent in distributed settings.

F2. Representational objects that can be exchanged and still be locally interpreted in ways that make collective sense, yielding semantic coherence to the actions of distributed algorithms.

This presents a kind of chicken-and-egg problem: distributed reasoning algorithms depend on collectively-sensible objects for coherent behavior, yet the collective sensibility of objects arises through distributed reasoning algorithms. So, our central questions are these:

Q1. How can we break into this cycle and create algorithms to bootstrap and adapt new collective object semantics over time?

Q2. How can we stabilize collective semantics in dynamic settings so that collective behavior is possible?

In short, we need distributed methods that manage both control uncertainty and semantic uncertainty in dynamic settings, and that reduce—or at least stabilize—both uncertainties over time. Two points of interactional and temporal space interest us: the long-term, large-scale dynamics of collective concepts and their overall structure, and the short term adjustment, adaptation, and innovation processes in which situation-specific concepts and interpretations emerge quickly within a context of the more stable and long-term conceptual architecture. We are led to viewing collective concept management as a multi-tiered dynamic system built from constrained, regionalized (i.e., not globally random) interactions, that may have local dynamism while exhibiting a degree of global stability. In our research work as a whole, we approach these two levels with two different methodologies: algorithms for rapid multi-agent mutual concept design and alignment for the local, short term adaptation, and a "population dynamics of concepts" for understanding, predicting, and stabilizing the longer-term structural properties of conceptual spaces. In this paper we treat only the design of specific algorithms for the short-term, context-sensitive mutual adaptation of ontologies by a collection of autonomous agents. Our work on the issues of stability and long-term structure of collective ontologies is treated elsewhere.

Agents cooperating in a multi-agent setting need a shared ontology[8]. But the fact is that there is often more than one ontology established in a given domain. Since these existing ontologies are designed by different people or organizations, *ontology alignment*[1] is needed for agents to cooperate in multi-agent systems. Recently there has been an increasing interest in ontology alignment [6, 4, 1, 5, 2].

However, existing work on ontology alignment does not fully accommodate requirements for open multi-agent systems. Open multi-agent systems must adapt rapidly to new categories, conceptualizations, and specifications of domains — they cannot be defined or aligned once and for all. Moreover, autonomous multi-agent systems must refine and align their own ontologies collectively, not through the external human intervention of standards committees and designers. To create multi-agent systems that are both adaptive and open, agents must collectively design common ontologies actively in an online fashion.

To this end, we propose a mutual ontology adaptation frame-

[1]By *ontology alignment* we mean a process that is often approached through ontology mapping and/or merging.

work. The framework assumes that each agent has its own ontology which is *not* open to other agents or outside. An agent learns about the ontology of other agents through the exchange of information. In this paper, we define an ontology as an *extensional* categorization of the domain of interest, and we define the information that agents exchange as an instance or object of an ontological category. We will describe this approach formally in next section. The intuition behind this assumption is that each person in human society has its own ontology or conceptualization about the world, and one way that we know about the ontologies of others is through the exchange (or reference to) of instances such as concrete examples rather than the exchange of an abstract concept.

Based on the above framework, we design an ontology alignment game in which agents with different ontologies mutually adapt to each other's ontology through interaction. We also conduct an experimental simulation of the game which shows how agents can converge to a shared ontology, using a mutual ontology update algorithm proposed in this paper.

2. ONTOLOGY ALIGNMENT PROBLEM FORMAL DESCRIPTION

An ontology is a system of categories for a domain [7]. Let there be n agents $\{A_1, \cdots, A_n\}$, and a domain O which has m instances or objects $O = \{o_1, \cdots, o_m\}$. Each agent A_i has a categorization C_i on the domain O. A categorization C is a partition of the domain O: $C = c_1 \cup c_2 \cup \cdots \cup c_k$, where $c_i = \{o_1, \cdots, c_{l_i}\}$ is called a category. For example, let a domain O have 10 instances: $O = \{0, 1, ..., 9\}$. A categorization can be: $C = \{0, 1, 2, 3\} \cup \{4, 5, 6, 7\} \cup \{8, 9\}$, where $c_1 = \{0, 1, 2, 3\}$, $c_2 = \{4, 5, 6, 7\}$, and $c_3 = \{8, 9\}$ are three categories.

A single category is a concept of an agent's ontology. Note that in this definition, a category is an *extensional* representation of a concept. Most of existing work on ontology assumes that the representation of a concept is *intensional* description.

Given the above definition, we can identify two different ontology alignment problems: off-line ontology alignment and online ontology alignment. Most existing work belongs to off-line alignment, in which the entire specification for the problem exists before the alignment is done.

The task of off-line alignment can be defined as follows. Find a categorization C^* such that $\sum_{i=1}^{n} \mathcal{M}(C^*, C_i)$ is minimal. The mis-alignment function $\mathcal{M}$ can be defined according to the needs of application. Off-line alignment requires that an agent knows the ontology of other agents.

In this paper, we are concerned about online ontology alignment problem, which is defined as follows. Design a mechanism or find a solution such that $\sum_{i,j=1}^{n} \mathcal{M}(C_i^{(t+1)}, C_j^{(t+1)}) < \sum_{i,j=1}^{n} \mathcal{M}(C_i^{(t)}, C_j^{(t)})$, where t means the time step. In online mode, we don't care about finding a global optimal categorization such as C^* in off-line mode. Since the alignment process in this setting is dynamical, and thus the result is adaptable as the situation changes. If we can work out a mechanism to satisfy the above inequality, it means all agents will eventually converge to a shared ontology.

3. MUTUAL ONTOLOGY ADAPTATION FRAMEWORK

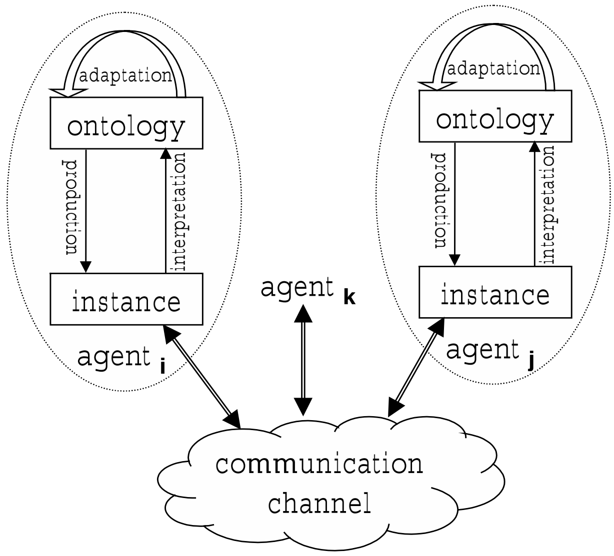

Figure 1: Framework of mutual ontology adaptation

To approach the online ontology alignment problem, we introduce a mutual online ontology adaptation framework (see figure 1). A mutual online ontology adaptation system consists of a set of agents and a communication channel used by agents to exchange information. In this paper we suppose the communication channel is perfect without noise. For one agent, the framework includes five elements: *ontology*, *instance*, *instance production*, *instance interpretation*, and *ontology adaptation* (or *online ontology alignment*).

Ontology
A partition of the domain O (see the previous section).

Instance
An element o in the domain O: $o \in O$.

Production
When an agent plans to express some concept or category to other agents, it can use an instance belonging to that category to represent this concept[2]. There might exist many instances that can be used to represent the concept. It is possible that different instances have different priorities to represent a same concept.

Interpretation
Given an instance, find the category of this instance.

Adaptation
When an agent receives an instance from another agent, it may adapt its ontology such that its ontology would be more compatible with the ontology of the sender. We will give a specification of some adaptation rules in the next section.

[2]In fact, unless a set of agents *already has* a compatible and verified shared ontology, it is difficult to see how they could specify categories to each other in another way.

4. ONTOLOGY ALIGNMENT GAME

We model the multi-agent mutual ontology alignment as an *ontology alignment game*. The game is played repeatedly in a (finite or infinite) sequence of rounds. On each round, the following events happen:

1. Two agents are randomly chosen from the agent population. One is called the *learner* A_L and the other is called the *teacher* A_T. In fact, this terminology is not completely accurate, since both agents will adapt their ontology in the same round, learning from each other. However, there is a slight asymmetry because the teacher will initiate the process. Denote by C_T the teacher's categorization or ontology, and by C_L the learner's one.
2. The teacher randomly chooses a category c from its ontology C_T, and then chooses an instance $o \in c$.
3. The learner receives the instance o, and performs an interpretation, locating a category $c' \in C_L$ such that $o \in c'$. From c', the learner chooses a new object o' from c' which is different from the received one, and sends o' back to the teacher to indicate that to the learner's knowledge c', o and o' all belong to the same concept or category.
4. The teacher tells the learner whether the instance o' actually does match the category c of the teacher or not. If it matches, we say that the learner has made a correct interpretation on this round.
5. The learner takes certain actions (e.g., updating its ontology or doing nothing), after getting the feedback from the teacher. The teacher also takes certain actions (e.g., updating its ontology or doing nothing) after sending the feedback to the learner. Below we detail these actions and the decisions that lead to them.

The goal of an agent in a learner's role is to make as few mistakes as possible during the game. If possible, we would like all agents to converge to a shared ontology so that no interpretation mistakes will be made in the population. One good measure of success is the probability that any agent makes no mistakes of interpretation.

4.1 Ontology update

In our model, we assume that each agent keeps an association matrix, which is used to capture relationships among objects in the domain of interest. After each round of the alignment game, the association matrix might be updated to reflect the new relationships among objects. Given a current version of an agent's ontology and a newly updated association matrix, a new version of ontology can be computed using a variant of K-means — a classical partition-based clustering method[3].

Association matrix update

Denote by s_{ij} or $s_{(o_i,o_j)}$ the association strength of the instances pair (o_i, o_j). Let $s^T_{(o,o')}$ be the association strength between the two objects o and o' on the teacher's side, and $s^L_{(o,o')}$ be the association strength between o and o' on the learner's side. If the learner's response matches the teacher's, the update will be:

$$\begin{cases} s^T_{(o,o')} & \leftarrow & s^T_{(o,o')} + \alpha \\ s^L_{(o,o')} & \leftarrow & s^L_{(o,o')} + \alpha \end{cases} \tag{1}$$

where α ($0 \leq \alpha < 1$) is called *match* update rate. When $\alpha = 0$, it means doing nothing. Teacher and learner can take different α values. For simplicity, we assume that they take the same value.

If the learner's response doesn't match the teacher's, the update will be:

$$\begin{cases} s^T_{(o,o')} & \leftarrow & s^T_{(o,o')} + \beta \\ s^L_{(o,o')} & \leftarrow & s^L_{(o,o')} - \beta \end{cases} \tag{2}$$

where β ($0 \leq \beta < 1$) is *mismatch* update rate. Similarly, when $\beta = 0$, it means doing nothing. Teacher and learner can take different β values. For simplicity, we assume that they take the same value.

Note that *both* learner and teacher will update their matrices. The intuition behind the joint update is that neither agent's existing ontology is treated as the ideal reference. In a sense, the learner has made a mistake because the teacher, too, has made a mistake: if the teacher's ontology had been corrected with respect to the learner (i.e. identical to the learner's ontology), then the learner would not have made a mistake. The collection of agents is collaboratively designing an ontology that fits their mutual circumstances, rather than moving toward a pre-ordained one.

Ontology update

Let an agent have a set of categories as its ontology: $C = c_1 \cup \cdots \cup c_n$. The agent also has an association matrix $S = (s_{ij})$. Initially, s_{ij} is set to:

$$s_{ij} = \begin{cases} 1 & \text{if object } o_i \text{ and } o_j \text{ belong to the same category} \\ 0 & \text{otherwise} \end{cases}$$

The update of categorization C is done by using a variant of K-means clustering. The basic idea is as follows. For an object o, compute the association strength between o and each category c_i, which can be defined as:

$$AS(o, c_i) = \frac{1}{\mid c_i \mid} \sum_{o' \in c_i} s_{(o,o')}$$

For each object, assign it to a category which has the maximal association strength with this object. The (re)assignment of object to a category may be repeated until some criterium is satisfied. In this paper, we only conduct once for object assignment.

5. SIMULATION

We have constructed a simple simulation that represents this collective ontology alignment. To make things simple, in this experiment, we consider only the categorization of fixed number of categories[3].

[3]In fact, K-means clustering requires the number of clusters be fixed in advance.

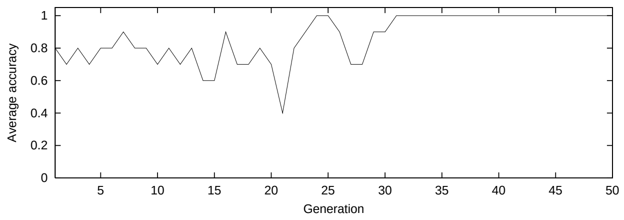

Figure 2: Simulation of ontology alignment game.

Figure 2 shows such a simulation of 2 agents in a domain containing 10 instances $\{0, 1, 2, \cdots, 9\}$ with 2 fixed categories. We set the two update rates as: $\alpha = \beta = 0.1$. Each generation contains 10 rounds of the ontology alignment game—one for each agent as teacher with a random learner (though the selection of agents doesn't make a large difference). The *average accuracy* is the ratio of the total number of correct interpretations over all rounds in the single generation to the number of rounds in that generation (i.e., 10 in this case). As the figure shows, the collection of agents in this simulation converges to a perfect collective interpretation at about 31 generations.

Table 1: The evolution of agents' ontologies.

Initial ontologies of agents		
agent 1	{0, 1, 2, 3, 4}	{5, 6, 7, 8, 9}
agent 2	{0, 2, 4, 6, 8}	{1, 3, 5, 7, 9}
Shared ontologies after evolution		
agent 1	{0, 2, 4}	{1, 3, 5, 6, 7, 8, 9}
agent 2	{0, 2, 4}	{1, 3, 5, 6, 7, 8, 9}

Table 1 illustrates what the evolution result looks like. After the convergence, they share the same ontology. According to the initial ontologies of two agents, we are able to figure out that there exist two optimal shared ontologies[4] : $C = \{0, 2, 4\} \cup \{1, 3, 5, 6, 7, 8, 9\}$ and $C = \{5, 7, 9\} \cup \{0, 1, 2, 3, 4, 6, 8\}$. This simulation shows the first solution. In fact, with different random seed setting, another solution was also obtained in our experiment.

6. DISCUSSION AND CONCLUSION

We have given the sketch of a formalized approach to a dynamic process for mutual continuous co-alignment (actually a kind of collective design) of a group-wide ontology. At present we have no formal characterization of the specific conditions for collective convergence of this procedure, or proof of convergence in all cases. However, our simulation work indicates that there is a wide range of conditions under which convergence does occur, and we believe a more rigorous result can be built. Further development of this idea will include improving our understanding of these convergence conditions, as well as expanding our treatment to cover more forms of ontological representation with greater structural complexity.

[4]Here, we define *optimal shared ontology* as the ontology which has the minimal mismatch with the initial ontologies of all agents.

7. ACKNOWLEDGMENTS

This work was partially supported under NSF grant 9981-797, and by the Information Systems Research Lab (ISRL) of the Graduate School of Library and Information Science. We thank the reviewers for helpful comments.

8. REFERENCES

[1] R. Agrawal and R. Srikant. On integrating catalogs. In *WWW2001*, Hong Kong, May 2001.

[2] A. Doan, J. Madhavan, P. Domingos, and A. Halevy. Learning to map between ontologies on the semantic web. In *WWW2002*, Honolulu, Hawaii, May 2002.

[3] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.

[4] M. S. Lacher and G. Groh. Facilitating the exchange of explicit knowledge through ontology mappings. In *The 14th International FLAIRS Conference*, Key West, FL, 2001. AAAI Press.

[5] A. Maedche and S. Staab. Ontology learning for the semantic web. *IEEE Intelligent Systems*, 16(2):72–79, March/April 2001.

[6] N. F. Noy and M. A. Musen. Prompt: Algorithm and tool for automated ontology merging and alignment. In *AAAI/IAAI 2000*, pages 450–455, Austin, TX, 2000.

[7] J. Sowa. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks/Cole, Pacific Grove, CA, 2000.

[8] L. Steels. The origins of ontologies and communication conventions in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 1(2):169–194, October 1998.