An Agent-Based Simulation Perspective for Learning/Merging Ontologies

Adrian Giurca[1] and Gerd Wagner[1]

[1]Brandenburgische Technische Universität, Germany
{Giurca, G.Wagner}@tu-cottbus.de

1 Introduction

Ontologies can be learned from various sources, be it databases, structured and unstructured (Web) documents or even existing preliminaries such as dictionaries and taxonomies. In addition, the distributed nature of ontology development has led to a large number of different ontologies covering the same or overlapping domains therefore the research community should deal with issues such as ontology mapping and merging too. This topic is addressed by the cognitive science community by means of language learning simulation. The problem of ontology learning overlaps with the one of language learning: both of them address the issues of learning from text, learning of concepts and taxonomies. Ontology mapping can be viewed also as a language learning process since it defines in fact a common vocabulary derived from the previous non-mapped vocabularies. Our proposal is to investigate the potential of an agent-based discrete event simulation framework to perform simulations resulting in language learning and evolution and consequently offering other solutions to the ontology learning and mapping problems and/or evaluating others solutions.

Individual learning is the knowledge acquired in every situation in which an agent reacts and processes data, including its beliefs about its actions in order to improve the performance in similar situations in the future. Such process aims to align the agent beliefs to the objective real world. Usually, in the initial state, the agents will have no common lexicon and therefore no understanding of what other agents say to them. The expectation is, that the agents will develop in time a shared vocabulary and ultimately a shared ontology (see [1] and [2]). Although agents start without any knowledge about the world, so that they have no representations of meaning, the goal is to have a population evolving a common language with which they can communicate.

A comprehensive classification of ontology learning approaches and tools before 2000 can be found in [3]. The term *ontology learning for the Semantic Web* was coined by Maedche and Staab [4] and largely addressed in [5]. They established a research direction and specified a first architecture for ontology learning. After that a number of tools were created. Significantly we see: AIBF, TextToOnto ([6], [7]), DFKI OntoLT ([8]), DFKI RelExt ([9]), but for sure there are many others. A good reference about all these works is [10]. Recently an *Ontology Learning Layer Cake* discussing learning of terms, synonyms, concepts, taxonomies, relations and axioms/rules was introduced (see [11]).

In the last ten years many researchers developed methodologies and tools for ontology mapping and ontology merging, critical operations for information exchange on the Semantic Web. A proposal for ontology mapping was introduced in 2004 ([12]). The work proposed to determine similarities through rules which have been encoded by ontology experts. A more theoretical work ([13]) proposed an algebraic solution to capture merging of ontologies by pushouts construction from category theory. They built this solution independent of a specific choice of ontology representation. Another solution was proposed by the GLUE system ([14]) who introduced a machine learning approach to find ontology mappings. Started in 2004, the Ontology Alignment Evaluation Initiative aims to describe a form of consensus with respect of (a) assessing strength and weakness of alignment/matching systems; (b) comparing performance of techniques, and (c) improve evaluation techniques, through the controlled experimental evaluation of the techniques performances. The initiative delivered an API for ontology alignment ([15] and recently a book was published [16].

2 An Agent-Based Discrete Event Simulation Framework

AOR Simulation provides an agent-based discrete event simulation framework (`http://aor-simulation.org`) based on a high-level rule-based simulation language (*AORSL*) and an abstract simulator architecture and execution model with a reference Java implementation. Its main concepts have been proposed in [17] and a Java-based simulation tool (AOR-JavaSim) has been developed.

A *simulation scenario* is expressed in the AOR Simulation Language (AORSL) and then Java source code is generated, compiled to Java byte code and finally executed. It consists of a *simulation model*, an *initial state of the world* and possibly *view definitions*. The simulation model consists of: (1) an optional *space model* (needed for physical objects/agents visualization); (2) a set of entity types, including event types, messages, objects and agent types; (3) a set of environment rules, which define causality laws governing the environment state changes. A simulation can use various *space models* characterized by: (i) dimension (1D, 2D or 3D); (ii) discrete/continuous and (iii) geometry (Euclidean or Toroidal).

An agent type is defined by means of: (1) a set of (objective) properties; (2) a set of (subjective) self-belief properties; (3) a set of (subjective) belief entity types; (4) a set of agent rules, which define the agent's reactive behavior in response to events and (5) an optional set of communication rules defining the agent-to-agent communication capabilities. Agent beliefs might be defined as *knowledge of the entity about it self and/or about the external world: objects, events or other agents*. Therefore an agent may have two types of beliefs (Figure 1): (1) self beliefs properties - knowledge of the agent about it self; (2) belief entities - knowledge of the agent about other agents, objects or events related to its world during a simulation. The upper level ontological categories of AOR Simulation are *messages*, *events* and *objects*. Objects include *agents*, *physical objects* and *physical agents*.

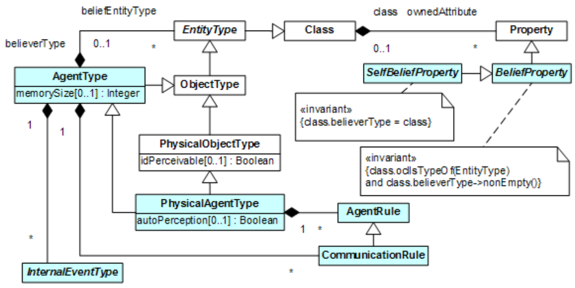

Fig. 1. Modeling Agents and Beliefs

The ontology of event types (see Figure 2): (a) environment events types (including *exogenous events types*, *perception event types* and *action event types*), and (b) internal events (such as *actual perception event types* and *periodic event types*) has been proven to be fundamental in AOR Simulation. Internal events are those events that happen "*in the mind*" of the agent. For modeling distorted perceptions, both a perception event type and the corresponding actual perception event type can be defined and related with each other via actual perception mapping rules. Both *the behavior of the environment* (its causality laws) and t*he*

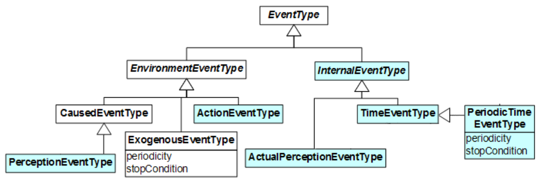

Fig. 2. Categories of event types.

behavior of agents are modeled with the help of rules, thus supporting high-level declarative behavior modeling.

AOR Simulation supports the *distinction between facts and beliefs*, including self-beliefs (the agent's beliefs about itself).

3 Research Opportunities

The typical AOR scenario for ontology learning and merging/mapping consists in a number of agent types, each of them having their own vocabulary about the real world. The agents interactions are the only way to communicate knowledge. A potential solution requires achievements on the following research questions:

1. AOR agents must be equipped with individual learning capabilities. However, there are several ways of implementing learning capabilities. *Which learning capabilities should offer AOR? Can we use just the machine learning community*

achievements as they are or specific solutions have to be considered? Looks like the standard individual learning can be implemented through Reinforcement Learning (RL), [18]. However, since the agents reasoning is encoded by means of rules the standard RL mechanics had to be adjusted accordingly. It seems that we will not use an explicit reward function based on a crisp optimization criterion. Our implicit reward does not reflect an objective function to be optimized (as in typical evolutionary algorithm applications), nor a concrete task to be performed optimally (as in evolutionary robotics). Our agents only need to survive and communicate in their environment (as in some ALife systems).

2. *Is the agent memory necessary? Is this related just to the remembering of the agents previous actions or it may be necessary a memory of its past beliefs too?* From the learning perspective, the agent needs a memory of its last experience for every action, where experience means a positive reward, negative reward or failed action. It, may need to remember all the perception events and messages that were present at the time step of that last experience. This enables agents to learn new mappings between state and actions by comparing previous experiences.

3. *What kind of reasoning capabilities are necessary for the agent?* Evolutionary learning and individual learning should both be performed by the agent reasoner. Hence, an agent can be created with a specific reasoner but change it during its lifetime by performing lifetime learning.

4 Conclusions

We have argued that the problem of merging ontologies by discovering ontology mappings might be also addressed by using an agent-based simulation based on existing literature, theories of learning, our experience, and an observational case study. In this position paper we developed a number of research questions that need to be investigated towards using cognitive science techniques to perform ontology learning and merging. The simulation results can be used by ontology engineers in the manual process of ontology learning/merging/refining or might be integrated in other tools for semi-automatic processing. From the main problem perspective, we see that the automated ontology learning/merging is a complex task. Based on our investigation, the problems users experience go beyond the processing of the algorithms. Users have to keep in mind what they have looked at and executed, to understand output from different algorithms, to be able to reverse their decisions, and to gather evidence to support their decisions. We believe that all these problems have to be addressed in an agent-based simulation and they constitute key assets for a successful solution.

We look towards other researchers feedback including ones which are interested to join our initiative.

References

1. Gopnik, A., Meltzoff, A.: Words, Thoughts, and Theories (Learning, Development, and Conceptual Change). Cambridge, MA, MIT Press (1997)

2. Vogt, P.: The emergence of compositional structures in perceptually grounded language games. Artificial Intelligence **167** (2005) 206–242
3. Maedche, A., Staab, S.: Learning ontologies for the Semantic Web. In: In Proceedings of the Second International Workshop on the Semantic Web. (2001) 200–210
4. Maedche, A., Staab, S.: Ontology learning for the semantic web. IEEE Intelligent Systems **16** (2001) 72–79
5. Maedche, A.: Ontology Learning for the Semantic Web. PhD thesis, Universität Karlsruhe (TH), Universität Karlsruhe (TH), Institut AIFB, D-76128 Karlsruhe (2001)
6. Maedche, A., Staab, S.: Ontology Learning from Text. In: Natural Language Processing and Information Systems, 5th International Conference on Applications of Natural Language to Information Systems, NLDB 2000. Volume 1959 of Lecture Notes in Computer Science., Springer (2000) 364
7. Cimiano, P., Völker, J.: Text2Onto. In: Natural Language Processing and Information Systems, 10th International Conference on Applications of Natural Language to Information Systems, NLDB 2005. Volume 3513 of Lecture Notes in Computer Science., Springer (2005) 227–238
8. Buitelaar, P., Olejnik, D., Sintek, M.: A protege plug-in for ontology extraction from text based on linguistic analysis. In: The Semantic Web: Research and Applications, First European Semantic Web Symposium, ESWS 2004. Volume 3053 of Lecture Notes in Computer Science., Springer (2004)
9. Schutz, A., Buitelaar, P.: RelExt: A Tool for Relation Extraction from Text in Ontology Extension. In: International Semantic Web Conference, ISWC 2005. Volume 3729 of Lecture Notes in Computer Science., Springer (2005) 593–606
10. Buitelaar, P., Cimiano, P., Magnini, B.: Ontology Learning from Text: Methods, Evaluation and Applications. Frontiers in Artificial Intelligence and Applications. IOS Press (2005)
11. Cimiano, P.: Ontology Learning and Population from Text. PhD thesis, Universität Karlsruhe (TH), Universität Karlsruhe (TH), Institut AIFB, D-76128 Karlsruhe (2006)
12. Ehrig, M., Sure, Y.: Ontology mapping - an integrated approach. In: The Semantic Web: Research and Applications, First European Semantic Web Symposium, ESWS 2004. Volume 3053 of Lecture Notes in Computer Science., Springer Verlag (2004) 76–91
13. Hitzler, P., Krötzsch, M., Ehrig, M., Sure, Y.: What is ontology merging? - a category theoretic perspective using pushouts. In: In Proc. First International Workshop on Contexts and Ontologies: Theory, Practice and Applications, AAAI Press (2005) 104–107
14. Doan, A., Madhavan, J., Dhamankar, R., Domingos, P., Halevy, A.Y.: Learning to match ontologies on the Semantic Web. VLDB Journal **12** (2003) 303–319
15. Euzenat, J.: An api for ontology alignment. In: The Semantic Web - ISWC 2004: Third International Semantic Web Conference. Volume 3298 of Lecture Notes in Computer Science., Springer (2004) 698–712
16. Euzenat, J., Shvaiko, P.: Ontology matching. Springer-Verlag, Heidelberg (DE) (2007)
17. Wagner, G.: AOR Modelling and Simulation - Towards a General Architecture for Agent-Based Discrete Event Simulation. In: Agent-Oriented Information Systems. Volume 3030 of LNAI. Springer-Verlag (2004) 174–188
18. Sutton, R., Barto, A.: Reinforcement Learning: An Introduction. Cambridge: MIT Press (1998)