

Training intelligent agents in the semantic web era: The golf advisor agent

Ioannis N. Athanasiadis Istituto Dalle Molle di Studi sull'Intelligenza Artificiale Lugano, Switzerland ioannis@idsia.ch

Abstract

Agent training techniques study methods to embed empirical, inductive knowledge representations into intelligent agents, in dynamic, recursive or semi-automated ways, expressed in forms that can be used for agent reasoning. This paper investigates how data-driven rule-sets can be transcribed into ontologies, and how semantic web technologies as OWL can be used for representing inductive systems for agent decision-making. The method presented avoids the transliteration of data-driven knowledge into conventional if-then-else systems, rather demonstrates how inferencing through description logics and Semantic Web inference engines can be incorporated into the training process of agents that manipulate categorical and/or numerical data.

1. Introduction

Agent training has been established as the procedure for periodically enhancing agent intelligence, and in particular has been investigated in relation to data mining techniques and inductive reasoning. The Agent Academy research project has coined the term "agent training" and provided with methods, procedures and tools for realising it in practice on software agents [9]. Through the Agent Academy approach it has been demonstrated how to transform datadriven knowledge into rule-bases that consequently are embedded into software agents and used for agent reasoning. In [12] we have presented a training methodology that transliterates induced knowledge into a rule base, defined in JESS and embedded into software agents using Agent Academy. This process has been demonstrated successfully in industrial applications as for supply chain management and environmental information processing [1]. Within the same line of efforts, we have investigated issues related to retraining [11], i.e the periodic update of agent's rule-base and the improvement in the efficiency of agent decision making. In all previous approaches have been based on the transliteration of inductive knowledge as rule bases, typically expressed in JESS [5]. In the meantime, there have been tremendous efforts on the development of the semantic web technologies, resulting to handy tools for reasoning using Description Logics and the Web Ontology Language (OWL) [6]. In this paper, we investigate how agent-training methodologies can be extended by using semantic web tools and how OWL ontologies can be utilized as representing agent reasoning that can be periodically updated and forming an ontology-based framework for agent training.

The rest of the paper is structured as follows: The following section 2 presents some background work on agent reasoning and training, and then in section 3 an ontology for training intelligent agents is introduced. Finally in Section 4 a golf-playing agent is presented as demonstration of the method.

2. Software agents and ontologies

Software agent and ontology research has so far concentrated in two aspects: agent communication and in agent reasoning. Typically, software agents employ ontologies for defining the semantics of agent communication, and FIPA¹-IEEE standards for agent communication provide with ontological foundations for agent messaging [3, 4]. These developments are approaching agency from an interoperability point of view and have been supported by pracctical tools for developing software agents, as $JADE^{2}[2]$ and deploying agent communication ontologies [13]. In a parallel effort, semantic web technologies have brought forth practical tools for reasoning with Description Logics, as for example Racer, which though was introduced as a "core OWL-reasoning agent for the semantic web" [7]. There are several efforts from the artificial intelligence and knowledge representation communities in deploying reasoning agents. However, software agent implementations that employ OWL ontologies is minimal, as the two approaches have been treated so far by different communities.

¹FIPA:Foundation of Intelligent Physical Agents

²JADE: Java Agent DEvelopment Framework

AgentOWL [8] was one of the first approaches working towards the combination of the two approaches, by providing with a framework for deploying OWL-based software agents with JADE. Similar is the situation with inductive reasoning agents, i.e. agents that do not incorporate deductive systems for their reasoning, rather they rended empirical decision-making models.

3 An ontology for training intelligent agents

An agent with state as defined by Wooldridge [14], as a process that percepts its environment, and based on its perception it updates its internal state, so that ultimately it may respond with some action. An agent a which is able to be trained, follows a similar model. Instead of a generic transformation function that turns perceptions into opinions, a is equipped with a rule base *rulebase*, which trsnforms agent percetions into their agent opinion, which in turn may result to agent actions.

In this respect, an inductive reasoning agent can be considered as a process that transforms the environmental states $s \in \mathbf{S}$ into perveptions through a function $see : \mathbf{S} \to \mathbf{P}$. Atomic or complex agent perceptions are subject to a rulebase, which turns them into an agent opinion $o \in \mathbf{O}$, as: $rulebase: \mathbf{P}^{\star} \to \mathbf{O}$. Finally agent opinions are converted into agent internal states and actions through the functions: $trans: \mathbf{I} \times \mathbf{O}^* \to \mathbf{I} \text{ and } action: \mathbf{I} \to \mathbf{A}.$

The rulebase that is introduced enables an agent to interprete its perceptions for shaping its opinion about the environmental conditions. Opinions may gradually turn into internal states and actions. The trans function could be linear, so that each opinion is transformed to a state, or could it be that it operates as a filter for identifying conditions for which the agent needs to change its internal state.

We also introduce here the ontology \mathcal{O} of agent a, which embraces all the resources that the agent is aware of. O pertains agent atomic and complex perceptions P^* , agent opinions O, agent internal states I and agent actions A. Therefore is $\mathcal{O} \supset \{P^* \cap O \cap I \cap A\}$. The agent training process train is the function that updates an agent's rulebase with a newer one. Agent ontology \mathcal{O}_a can be specified using the web ontology language OWL. In this case, agent perceptions and opinions can be defined as OWL Classes and agent rulebase can be defined as OWL Axioms that classify perceptions to agent opinions. Through an OWL implementation, the agent training corresponds into updating the axioms of agent's OWL ontology, and can be communicated trough agent messaging, following some authorization process.

4 A simple example: The golf advisor agent

For demonstrating agent training and reasoning using OWL, the common "play golf" data mining problem, introduced by Quinlan [10] is considered. In the "play golf" example, an inductive system is built for relating weather conditions to playing or not golf. The following table 4 presents the playing golf dataset, and Fig. 5 illustrates the empirically obtained decision tree using C4.5 algorithm for decision tree induction.

Lets build a golf-expert agent a_g , able to suggest whether to play golf or not, based on the induced decision tree from the avaiable dataset . Following the method described in the previous section, an OWL ontology \mathcal{O}_q may be defined for specifying agent a_g resources. Based on \mathcal{O}_g ontology, we then demonstrate the training process for constructing a_q rule base as a set of OWL axioms defined in \mathcal{O}_q .

Та	ble	1.	The	playing	golf	d	ata	set
		-	•					-

#	Outlook	Temp.	Hum.	Windy	Play Golf					
1	sunny	hot	high	false	don't play					
2	sunny	hot	high	true	don't play					
3	overcast	hot	high	false	play					
4	rainy	mild	normal	false	play					
5	rainy	cool	normal	false	play					
6	rainy	cool	normal	true	don't play					
7	overcast	cool	normal	true	play					
8	sunny	mild	high	false	don't play					
9	sunny	cool	normal	false	play					
10	rainy	mild	normal	false	play					
11	sunny	mild	normal	true	play					
12	overcast	mild	high	true	play					
13	overcast	hot	normal	false	play					
14	rainy	mild	high	true	don't play					

Agent a_g is capable for percepting the weather conditions, so its atomic percetions are:

AtomicPerception \supset {Outlook, Temperature, Humidity, Windy $\},\$

where:

 $Outlook \equiv \{Sunny, Overcast, Rainy\},\$

Temperature \equiv {*Hot, Mild, Cool*},

Humidity \equiv {Normal, High}, and

Windy \equiv {*True*, *False*}.

Agent a_a may read complex weather perceptions specified as collections of atomic perceptions:

Weather $\equiv \{x | \exists x.outlook \in Outlook \land x.temperature \in$ *Temperature* $\land x.humidity \in$ textitHumidity $\land x.windy \in$ Windy }

Agent ontology \mathcal{O}_g for the golf-advisor agent contains also the agent opinions O on playing or not golf: $GolfAdvice \equiv \{Play, DontPlay\} \subset O.$

The following Fig. 1 illustrates (on the left) a generic ontology for specifying inductive reasoning agents and parts of the golf advisor agent ontology³. Note that the golf advisor agent ontology has been developed using OWL.

The golf advisor agent a_g employed with the \mathcal{O}_g ontology can communicate data of the *Weather* class as instances in OWL. Data of table 4 can be serialized in OWL format as the following example for the first item:

```
<Weather rdf:ID="weather_01">
```

```
<Hot rdf:ID="hot_01"/>
<False rdf:ID="false_01"/>
<Sunny rdf:ID="sunny_01"/>
```



Figure 1. The inductive agent ontology (left) and partial view of the golf advisor agent ontology (right)

5 Golf advisor agent training

Having the golf advisor agent deployed with the \mathcal{O}_g ontology, it is aware of the semantics of the golf advising domain. What remains is the training of agent a_g so that its rule base is shaped. This may be realised by training from historical data, and can be periodically updated as new data become available. In the followings, we present the process of training agent a_g by means of specifying its rule base as OWL axioms.

Consider that at some point in time, only the first three rows of the data set in Table 4 are available for agent training. By using the C4.5 algorithm for decision tree induction, one may derive to a simple decision tree illustrated in Figure **??**.

This tree consists of two rules, that can be transcribed as axioms in \mathcal{O}_g ontology, as follows:

 $Rule1 \equiv \{\forall x | \exists x.outlook \in Overcast\} \subset Play$ $Rule2 \equiv \{\forall x | \exists x.outlook \in Sunny\} \subset DontPlay$ The agent training process involves the communication

of the rule set to agent a_g and its execution by the agent. In OWL notation, Rule1 can be written as:

```
<owl:Class rdf:ID="Rule1">
<owl:equivalentClass>
<owl:Restriction>
<owl:onProperty rdf:resource="#outlook"/>
<owl:someValuesFrom rdf:resource="#Overcast"/>
</owl:Restriction>
</owl:equivalentClass>
<rdfs:subClassOf rdf:resource="#Play"/>
</owl:Class>
```

At a later stage, lets assume that more data become available for agent training (actually the whole training set of Table 4). Then, by using the same algorithm for training, the decision tree of Figure 5 can be produced, that can be written as OWL axioms shown in Table 2. Agent retraining from an external point if view involves the induction of a newer decision tree, its transcription in OWL and its communication to the agent a_g . From an internal view, as the agent receives the new or updated axioms, it needs to update its rulebase with the new axioms.



Figure 2. The induced play golf decision tree, if trained with the first three records subset

 $^{^{3}}The golf advisor agent ontology is available online at http://www.idsia.ch/~ioannis/golfadvisoragent/$

Table 2. The golf playing agent rule set

 $\begin{aligned} Rule1 &\equiv \{\forall x | \exists x.outlook \in Rainy \land \exists x.windy \in True\} \subset Play \\ Rule2 &\equiv \{\forall x | \exists x.outlook \in Rainy \land \exists x.windy \in False\} \subset DontPlay \\ Rule3 &\equiv \{\forall x | \exists x.outlook \in Overcast\} \subset DontPlay \\ Rule4 &\equiv \{\forall x | \exists x.outlook \in Sunny \land \exists x.humidity \in High\} \subset DontPlay \\ Rule5 &\equiv \{\forall x | \exists x.outlook \in Sunny \land \exists x.humidity \in Normal\} \subset Play \end{aligned}$

6 Discussion

This paper presented how rich semantics, using OWL, can be used for defining both agent resources and reasoning. An ontology for agent training has been introduced and its use was demonstrated for decision-making. It was shown how OWL notation can be used for defining agent resources, but most importantly how it can be used for translating inductive rule sets into axioms for agent reasoning. The simple case of a golf advisor agent has been used as a test-case. The main benefit of the method is that rich semantics are employed uniformly for all agent resources. Aspects of agent communication, reasoning, training and retraining are treated homogeneously as they declaratively founded on top of the same ontological grounds.

The main limitation of the approach is that OWL reasoning is based on description logics. While OWL-DL is directly usable when agent resources are nominal, as for the golf advisor agent, in the most popular case of an agent required to reason on non-categorical data, then a mapping function needs to be incorporated in the agent function *see*, which assigns numerical resources to OWL classes. The rest of the approach will remain the same.

Finally, the agent training life-cycle is not reduced only into transcribing rules as ontological axioms. However, with the method and demonstration presented here, it was made clear that agent reasoning, either a result of a deductive or of an inductive process, it can be treated uniformly. Thus, agent training and retraining processes can be used for both inductive and deductive systems.



Figure 3. The induced play golf decision tree, result of training with all data

References

- I. N. Athanasiadis and P. A. Mitkas. An agent-based intelligent environmental monitoring system. *Management of Environmental Quality*, 15(3):238–249, 2004.
- [2] F. Bellifemine, G. Caire, A. Poggi, and G. Rimassa. JADE-A white paper. *EXP in search of innovation*, 3(3):6–19, September 2003.
- [3] FIPA. Ontology Service Specification. Doc. No. XC00086 D, Foundation of Physical Intelligent Agents, Geneva, Switzerland, 2001.
- [4] FIPA. ACL Message Structure Specification. Doc. No. SC00061 G, Foundation of Physical Intelligent Agents, Geneva, Switzerland, 2002.
- [5] E. J. Friedman-Hill. Jess in Action. Manning Publications, 2003.
- [6] D. L. M. Guinness, F. van Harmelen, et al. OWL Web Ontology Language overview. W3C Recommendation, W3C, 2004. www.w3.org/TR/owl-features/.
- [7] V. Haarslev and R. Moller. Racer: An owl reasoning agent for the semantic web. In *In Proc. of the Intl Workshop on Applications, Products and Services of Web-based Support Systems, IEEE/WIC Intl Conf on Web Intelligence*, pages 91–96, Halifax, Canada, 2003.
- [8] M. Laclavík, Z. Balogh, M. Babík, and L. Hluchý. AgentOWL: Semantic knowledge model and agent architecture. *Computing and Informatics*, 25:419–437, 2006.
- [9] P. A. Mitkas, D. Kehagias, A. L. Symeonidis, and I. N. Athanasiadis. A framework for constructing multi-agent applications and training intelligent agents. *Lecture Notes in Computer Science (Agent Oriented Software Engineering IV)*, 2935:96–109, Jan 2004.
- [10] J. R. Quinlan. C4.5 Programs for Machine Learning. Morgan Kaufmann, 1993.
- [11] A. L. Symeonidis, I. N. Athanasiadis, and P. A. Mitkas. A retraining methodology for enhancing agent intelligence. *Knowledge-based Systems*, 20(4):388–396, 2007.
- [12] A. L. Symeonidis, K. C. Chatzidimitriou, I. N. Athanasiadis, and P. A. Mitkas. Data mining for agent reasoning: a synergy for training intelligent agents. *Engineering Applications of Artificial Intelligence*, (accepted for publication), 2007.
- [13] C. van Aart, R. Pels, G. Caire, and F. Bergenti. Creating and Using Ontologies in Agent Communication. In Ontologies in Agent Systems, 1st Intl Conf on Autonomous Agents and Multi-Agent Systems, Bologna, Italy, 2002.
- [14] M. Wooldridge. Intelligent Agents. In G. Weiss, editor, *Multiagent Systems: A modern approach to distributed Artificial Intelligence*, chapter 1, pages 27–78. MIT Press, 2000.