# **Mixed-Initiative Ontology Learning**

Cristina Boicu, Gheorghe Tecuci

Learning Agents Laboratory, Department of Computer Science, MSN 4A5, George Mason University, 4400 University Dr., Fairfax, VA 22030-4444, USA {ccascava, tecuci}@gmu.edu; http://lalab.gmu.edu; (703)993-4669; (703)993-1722

**Abstract.** This paper presents a mixed-initiative assistant that supports a subject matter expert to extend the ontology of a learning agent, in order to express the subtle distinctions he makes in an application domain. This assistant performs a comparative analysis of the correct examples and the exceptions accumulated by the problem solving rules and guides the expert in improving its knowledge base. This assistant is integrated into the Disciple-RKF system, and has been evaluated in two knowledge acquisition experiments performed at the US Army War College.

Keywords: Knowledge acquisition, machine learning, ontology learning, subject matter expert

# **1** Introduction

In the past years, the Defense Advanced Research Projects Agency (DARPA) investigated the rapid development and application of large knowledge bases under several programs. One of them was the Rapid Knowledge Formation (RKF) Program (Burke, 1999), whose main goal was to enable distributed teams of subject matter experts with no prior knowledge engineering experience to directly develop knowledge bases for complex applications.

A successful approach developed in this program relies on a very capable learning and reasoning agent called Disciple-RKF (Tecuci et. al 2002) that collaborates with a subject matter expert to develop its knowledge base, consisting of an object ontology and a set of general problem solving rules. In this approach, a knowledge engineer works first with the expert to define the object ontology of Disciple. This ontology consists of hierarchical descriptions of objects and features from the application domain. Then, the expert teaches Disciple to solve problems in a similar way to how the expert would teach a student, by giving the agent examples and explanations, as well as by supervising and correcting its behavior, when it attempts to solve new problems. During these interactions, the agent learns general problem solving rules by generalizing the individual examples and their explanations provided by the expert. One of the main difficulties in this process is that the agent's ontology is generally incomplete for any complex real-world domain. Therefore, the agent's knowledge base needs to be continuously extended to represent better the expertise in the application domain, capturing the subtleties of the expert's problem solving knowledge.

In the next two sections we will illustrate a mixed-initiative assistant for learning new elements to extend the object ontology, based on the exceptions accumulated by the problem solving rules. In section 4 we will present evaluation results of this assistant obtained as part of two knowledge acquisition experiments performed at the US Army War College in Spring 2002 and Spring 2003. We will conclude the paper with a presentation of related research and final remarks.

## **2** Rules with Exceptions

The object ontology plays a critical role in the learning and teaching process of the Disciple agent, being used as the generalization hierarchy for learning. However, this ontology is generally incomplete, causing situations in which there is no represented knowledge to distinguish between correct and incorrect problem solving examples. In such cases, trying to specialize a rule in order to uncover the incorrect examples will result in uncovering some of the correct examples as well. Therefore, in the context of the current ontology, the incorrect examples that could not be distinguished from the positive examples are kept as negative exceptions of the rule.

In this paper we present a mixed-initiative assistant for learning new object features that extend the object ontology, allowing the elimination of the rule's exceptions and improving the agent's problem solving efficiency. To present the assistant we will consider an example from a didactic domain, the PhD advising domain, which was used in a knowledge-base development experiment performed as part of the "IT 803 Instructable Agents" course at George Mason University in Spring 2002. During this course, the students developed a Disciple agent for assessing whether a faculty member would be an appropriate advisor for a specific student, considering the professor's area of expertise versus the student's research interests. The agent's goal was to identify the strengths and weaknesses of potential advisors with respect to several characteristics, thus being able to assist a person in choosing a PhD advisor. The students, acting as subject matter experts, taught personal Disciple agents to analyze potential PhD thesis advisors with respect to characteristics such as professional reputation, responsiveness to students, expected learning experience, financial support offered to students, general personality and work compatibility, and the expected quality of student's work.

The problem solving paradigm of Disciple is task reduction, which was used by the experts to express their reasoning, by decomposing an initial task into simpler tasks, guided by questions and answers, until a solution was found. In this teaching process, the experts formulated first in natural language specific problem solving examples, based on which Disciple learned general task reduction rules, through a mixed-initiative interaction.

For instance, an example of a problem solving step is:

- - -

If the task is to:			
Determine whether Dan_Smith is likely to be available for the duration of the dissertation of Bob_Evens.			
<i>I</i> ask the question: Does Dan_Smith has a long_term_position?			
The answer is: Yes, a tenured_position.			
Therefore I conclude that:			
Dan_Smith is likely to remain on the faculty of George_Mason_University for the duration of the dissertation of			
Bob_Evens because he has a tenured_position.			

From this specific problem solving example, Disciple learns a general task reduction rule, which is shown in Figure 1. The learned rule contains both an informal and a formal description. The informal description is the generalization of the initial example, as shown in the upper part of Figure 1. The formal description, shown in the bottom part of Figure 1, is

used in the internal reasoning of the agent and contains two applicability conditions: a plausible upper bound (PUB) condition and a plausible lower bound (PLB) condition, which approximate the exact applicability condition of the rule (Tecuci, 1998). This rule will be applied by the agent to solve new problems and the critique received from the user will be used to refine it.

For example, this rule will be applied in another situation in which the agent is inferring that professor Henry White will be available as a PhD advisor for Gina Davis. This generated problem solving step is shown in the left part of Figure 2. However, this example is wrong, because even though Henry White is a tenured professor, he actually plans to move to Carnegie Mellon University. Therefore, Henry White will not be available for the duration of Gina Davis' dissertation.

However, the current ontology of the Disciple agent does not contain any relevant element to distinguish between professors "Dan Smith" (from the positive example from which the rule was learned), and "Henry White" (from the incorrect example). More specifically, the ontology does not represent the fact that Henry White plans to move to Carnegie Mellon University, which is the differentiating element between Dan Smith and Henry White. Therefore, this negative example will be kept as a negative exception of the rule. Such exceptions decrease the learning and problem solving efficiency and indicate potential problems in the knowledge base. However, the exceptions may also suggest what is missing or partially represented in the agent's knowledge base, and may provide valuable information on how the ontology should be extended to represent the subtle

# Determine whether ?O1 likely to be available for the duration of the dissertation of ?O2.

**Question:** Does ?O1 has a long\_term\_position? **Answer:** Yes, a ?O3.

#### THEN

?O1 is likely to remain on the faculty of ?O4 for the duration of the dissertation of ?O2 because he has a ?O3.

#### IF

IF

Determine whether a faculty member is likely to be available for the duration of the dissertation of a student. The faculty member is ?O1 The student is ?O2

	PUB Condition ?O1 is faculty_member has_as_position ?O3 has_as_employer ?O4	PLB Condition ?O1 is professor has_as_position ?O3 has_as_employer ?O4
	?O2 is person	?O2 is PhD_student
	?O3 is long_term_position	?O3 is long_term_position
	?O4 is employer	?O4 is university
THEN A faculty member is likely to remain on the faculty of an university for the duration of the dissertation of a studer because he has a a long term position. The faculty member is ?O1 The student is ?O2 The long term position is ?O3		
	The university is ?04	

Figure 1: A learned rule

distinctions that experts make in their domain, as illustrated in the next section.

# **3** The Mixed-Initiative Exception-Based Learning Assistant

To address such problems, we have developed an Exception-Based Learning Assistant that interacts with the subject matter expert to comparatively analyze the negative exceptions and the positive examples of a rule, in order to discover extensions to the ontology (such as new features and/or new facts of the form *object feature value*) that will eliminate the exceptions of the rule (Boicu, 2002). The Exception-Based Learning process consists of the following phases:

- 1. **Object selection:** the expert analyzes the objects from the negative exceptions and the positive examples of the rule and selects a set of objects (corresponding to a variable from the rule) to differentiate them;
- 2. Candidate discovery: the agent discovers the most plausible types of distinctions between the selected objects, which may reduce or eliminate the exceptions;
- 3. Candidate selection: the expert interacts with the agent to select one of the proposed distinction candidates;
- 4. **Ontology learning:** based on the selected candidate, the agent elicits from the expert the specific distinction between the objects and extends the ontology;
- 5. **Rule refinement:** the agent refines the applicability condition of the rule and eliminates the rule's exceptions based on the performed ontology extension.

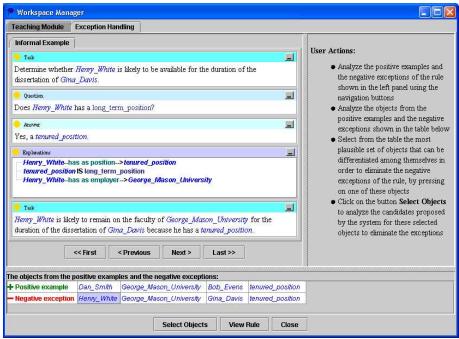


Figure 2: The Exception-Based Learning Assistant showing the negative exception

# The Object Selection Phase

Figure 2 shows the interface of the Exception-Based Learning Assistant in the object selection phase. In the right upper panel the agent displays a summary of the operations that need to be performed by the user. In the left upper panel the user can view the description of the rule's negative exception corresponding to "Henry White," expressed in natural language. The example is characterized by a set of objects (instances) from the agent's ontology (i.e. "Henry White", "George Mason University", "Gina Davis" and "tenured position") which are shown in the bottom panel, as a succinct representation of the example. The bottom panel in Figure 2 shows also a succinct representation of the positive example of the rule from Figure 1 (i.e. "Dan Smith", "George Mason University", "Bob Evens" and "tenured position"). This tabular representation of the objects from the positive examples and negative exceptions has the advantage of making easier the comparison between the examples, each column in this table representing a possible set of objects to be differentiated. This phase focuses the attention of the user on a specific set of objects to be differentiated, simplifying significantly the interaction and the ontology learning process. In order to distinguish between the positive example and the negative exception, the user selects the objects "Dan Smith" (from the positive example) and "Henry White" (from the negative exception) to express the difference between them.

#### The Candidate Discovery Phase

After the expert selected the objects "Dan Smith" and "Henry White" to differentiate them, the Disciple agent generates an ordered set of ontology extension candidates for these objects that have the potential of removing the exceptions, by using several heuristics to perform plausible reasoning and knowledge-base analysis (Boicu, 2003). The agent searches in the ontology for pieces of knowledge that distinguish between "Dan Smith" and "Henry White." For instance, the agent looks for features that characterize "Dan Smith" from the positive example, and which does not describe "Henry

White" from the negative exception. To each of the candidates that are automatically determined by the agent is associated a plausibility of eliminating the negative exception, which is used by the agent to order the discovered candidates.

#### **The Candidate Selection Phase**

After the candidates are generated by the Disciple agent, they are presented to the user for analysis. Figure 3 shows a fragment of the Exception-Based Learning Assistant's interface in the candidate selection phase. In the upper panel is shown a summary of the ontology extension candidates proposed by the agent, to differentiate between "Dan Smith" and "Henry White." The user may browse through the list of the candidates, and the agent will display in the bottom panel a detailed description of the currently analyzed candidate.



Figure 3: Proposed ontology extension candidates

For example, the agent determines that "Dan Smith" is the director of the "Artificial Intelligence Laboratory," while "Henry White" is not a director of any research laboratory or center. Because this feature may differentiate between "Dan Smith" and "Henry White," the agent proposes this candidate to the user, to analyze it. However, the fact that "Dan Smith" is the director of the "Artificial Intelligence Laboratory" does not capture the actual differentiating piece of information in this context (i.e. "Henry White" plans to leave to "Carnegie Melon University," while "Dan Smith" does not have such plans).

### The Ontology Learning Phase

Studying the proposed candidates, the user realizes that this piece of information is not captured in the agent's ontology, and therefore selects to define a new feature to distinguish between them. The user wants to define that the distinguishing feature is that Henry White plans to move to Carnegie Melon University and thus he will not be available for the duration of Gina Davis' dissertation.

The elicitation dialog for this new feature is shown in Figure 4. The user is guided by the agent to provide first the name of a new feature to express the difference between Dan Smith and Henry White. Then the expert is guided to define the differentiating values of the new feature for all the applicable objects. The user creates the new feature *"plans to move to"* and specifies that its value for "Henry White" is "Carnegie Melon University." In this context, "Dan Smith" does not plan to leave to another university and no value for this feature is associated to him.

Using the knowledge elicited from the expert, the agent learns a general definition of the feature "*plans to move to*" and refines the ontology to incorporate this knowledge. This definition includes the domain and the range of the feature, expressed as plausible version spaces, which will be refined as new examples are encountered.



Figure 4: Elicitation of a new feature

#### **The Rule Refinement Phase**

Based on this knowledge acquired from the user, the Disciple agent refines the reasoning rule from Figure 1, by adding an except-when condition shown in Figure 5. The except-when condition contains a plausible upper bound condition and a plausible lower bound condition, which will be refined as new examples are encountered. Therefore, the rule will no longer be applied in the cases in which the professor plans to leave at another university.

The refined rule does not cover any more the negative exception, which is transformed into a negative example of the rule, because now the agent knows why it is an incorrect problem solving episode. This process is successively repeated with each reasoning rule that contains exceptions, resulting in a more complete ontology and a set of refined problem solving rules with no or fewer exceptions.

EXCEPT-WHEN CONDITION:		
PUB Condition	PLB Condition	
?O1 is object	?O1 is associate_professor	
plans_to_move_to ?O5	plans_to_move_to ?O5	
?O5 is object	?O5 is university	

Figure 5: Added condition to the rule from Figure 1

# **4 Evaluation Results**

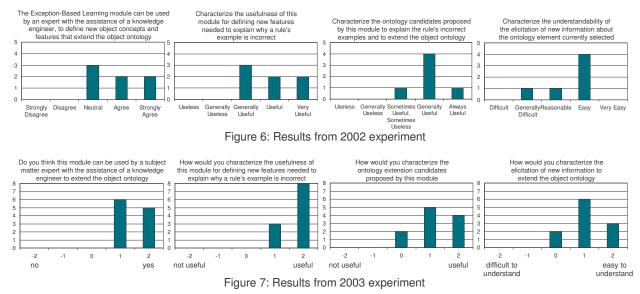
The Exception-Based Learning assistant was evaluated during two knowledge acquisition experiments performed with Disciple-RKF at the US Army War College, as part of the Military Applications of Artificial Intelligence (MAAI) course, taught in Spring 2002 and Spring 2003. This course was attended by 15 colonels and lieutenant colonels from different military services in Spring 2002, and 13 in Spring 2003. The students, who did not have prior knowledge engineering experience, were introduced to the Disciple approach, and used Disciple to develop an agent for the determination of the centers of gravity of the opposing forces from a war scenario. The concept of center of gravity is fundamental to military strategy, representing the primary source of moral or physical strength, power or resistance of a force (Strange, 1996). The main objectives of a force are to protect its own center of gravity, while attacking the one of the enemy.

During the experiment performed in Spring 2002, 7 subject matter experts used the Exception-Based Learning module with the assistance of a knowledge engineer, to eliminate the negative exceptions of the learned rules. During the training process of the agents, 8 problem solving rules accumulated 11 negative exceptions. In order to eliminate these exceptions, 4 new features were created and 6 feature-values were added, to express the difference between the positive examples and the negative exceptions. Each rule was refined with a new explanation, and its applicability condition was correspondingly updated.

During the experiment performed in Spring 2003, the experts used the Exception-Based Learning Assistant presented in this paper, which was an improved version of the previous module, assuring a more natural interaction, offering help and guidance to the expert through the entire process, and having better delimited phases. In this experiment, 11 subject matter experts grouped into teams of 2-3 members, used the Exception-Based Learning Assistant with the support of a

knowledge engineer, to eliminate the negative exceptions of the learned rules. During this knowledge-base development experiment, 7 of the learned problem solving rules accumulated 7 negative exceptions, indicating that some objects were not completely represented in the agent's ontology. As a result, 7 new features were created and 7 feature-values were added. The rules were correspondingly refined using this acquired knowledge.

Figure 6 and Figure 7 illustrate comparatively the assessments of the subject matter experts after using the Exception-Based Learning module in Spring 2002 and Spring 2003, respectively. There is a difference in the scale used: in 2002 we used a nominal scale, while in 2003 we adopted a numerical, equal interval scale from -2 to 2. Even though there are some small differences in the formulation of the questions, one can observe an improvement in the results obtained in the 2003 experiment comparing with the previous one. We consider this is the result of the improved version of the module, aimed to be easier to use by the subject matter experts.



As a general assessment of using the Exception-Based Learning Assistant, in the Spring 2003 experiment, the experts were also asked to answer the question *"How would you characterize the importance of using this module for refining the knowledge base"* on the scale from -2 (not important) to 2 (important). To this assessment, 10 subject matter experts answered "2" and one answered "1", indicating the validity of our approach.

# **5 Related Research and Final Remarks**

The Exception-Based Learning method is an improvement of the Consistency-Driven Feature Elicitation method (Tecuci and Hieb 1994; Tecuci 1998). The Exception-Based Learning assistant proposes several ontology extension candidates based on their plausibility to eliminate the exceptions, instead of a single extension suggested by the system. The Exception-Based Learning assistant is expert-oriented, instead of knowledge engineer oriented. Moreover, the Exception-Based Learning assistant considers ontology extensions for a subset of the positive examples and the negative exceptions and it allows the expert to choose the set of objects to be differentiated and the ontology extension to be performed.

The problem of extending the knowledge base to eliminate the exceptions of the learned rules is also addressed in the Mobal system (Morik et al. 1993; Wrobel 1994). The system contains a concept formation tool that supports a knowledge engineer in revising a knowledge base, by learning a concept definition that discriminates between the covered positive examples and all the exceptions of the rule. This concept is presented to the knowledge engineer, who may accept, modify or reject the system's proposal. An advantage of our method is that it can define several features to distinguish more naturally the positive examples from the negative exceptions. Also, in our approach the subject matter expert plays a key role in the ontology extension process, while Mobal's concept formation tool is oriented toward a knowledge engineer.

Another related approach is the repertory grid elicitation (Gaines and Shaw, 1993), in which the expert interacts with the system to define the relevant elements that will be analyzed from the application domain, and to describe the attributes (constructs) which distinguish these elements, using the triad method: the elements are presented in groups of three, the expert is asked to name an attribute that two elements share and the third one does not and finally both similarity and difference poles of the construct are named. This method is used in an initial process of acquiring knowledge about the application domain, while the Exception-Based Learning assistant in used in a problem solving context, to refine the existing knowledge base of the agent. In our approach the user is not restricted in its comparative analysis to only three objects; the new features defined by the user may have multiple values (not only bipolar values as

required in the repertory grid elicitation); and it allows the definition of several features that distinguish between the positive examples and negative exceptions.

In conclusion, our experiments showed that the Exception-Based Learning Assistant can be used to learn new ontological knowledge that represents better the subtle distinctions from an application domain, allowing the elimination of the rule's exceptions. It also results in improving the accuracy of the learned problem solving rules, by refining their plausible version space conditions. Moreover, the agent's problem solving efficiency is also improved by eliminating the need to explicitly check the exceptions each time when trying to apply a rule.

We plan to improve the Exception-Based Learning Assistant by allowing the learning of new object concepts to eliminate the rules' exceptions and by using analogical reasoning and hints from the user in the discovery of ontology extension candidates.

Acknowledgments: This research was sponsored by DARPA, AFRL, AFMC, USAF, under agreement number F30602-00-2-0546, by the AFOSR under grant no. F49620-00-1-0072, and by the US Army War College. The authors want to thank Mihai Boicu for his very useful comments and suggestions. Dorin Marcu developed the interface of the first version of this module, Bogdan Stanescu, Marcel Barbulescu and other members of the LALAB contributed to successive versions of Disciple.

#### References

Cristina Boicu, Gheorghe Tecuci, Mihai Boicu. 2002. Mixed-Initiative Exception-Based Learning for Knowledge Base Refinement. In *Proceedings of the Eighteenth National Conference of Artificial Intelligence*. Edmonton, Alberta, Canada.

Cristina Boicu, Gheorghe Tecuci, Mihai Boicu, Dorin Marcu. 2003. Improving the Representation Space through Exception-Based Learning. In *Proceedings of the Sixteenth International FLAIRS Conference* (Florida Artificial Intelligence Research Society). Saint Augustine, Florida.

Murray Burke. 1999. Rapid Knowledge Formation Program Description at http://reliant.teknowledge.com/RKF/projects /Darpa\_RKF\_PIP.htm

Brian Gaines and Mildred Shaw. 1993. Knowledge acquisition tools based on Personal Construct Psychology. *Knowledge Engineering Review*. 8(1), 49-85.

Katharina Morik, Stefan Wrobel, Jorg-Uwe Kietz, and Werner Emde. 1993. Knowledge Acquisition and Machine Learning – Theory, Methods and Applications. London: Academic Press.

Bogdan Stanescu, Cristina Boicu, Gabriel Balan, Marcel Barbulescu, Mihai Boicu, Gheorghe Tecuci. 2003. Ontologies for Learning Agents: Problems, Solutions and Directions. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI 2003) Workshop on Ontologies and Distributed Systems*. Acapulco, Mexico.

Joe Strange. 1996. Centers of Gravity & Critical Vulnerabilities: Building on the Clausewitzian Foundation So That We Can All Speak the Same Language. Quantico, VA: Marine Corps Univ.

Gheorghe Tecuci and Michael Hieb. 1994. Consistency-driven Knowledge Elicitation: Using a Learning-oriented Knowledge Representation that Supports Knowledge Elicitation in NeoDISCIPLE. *Knowledge Acquisition Journal* 6(1):23-46.

Gheorghe Tecuci. 1998. Building Intelligent Agents: An Apprenticeship Multistrategy Learning Theory, Methodology, Tool and Case Studies. London, England: Academic Press.

Gheorghe Tecuci, Mihai Boicu, Dorin Marcu, Bogdan Stanescu, Cristina Boicu, Jerry Comello. 2002. Training and Using Disciple Agents: A Case Study in the Military Center of Gravity Analysis Domain. *AI Magazine*. 23(4): 51–68.

Stefan Wrobel. 1994. Concept Formation and Knowledge Revision. Dordrecht, Netherlands: Kluwer Academic Publishers.