In *Proceedings of the Technology Demonstration Session of the 14th International Conference on Knowledge Engineering and Knowledge Management,* EKAW 2004, 5-8th October 2004 - Whittlebury Hall, Northamptonshire, UK.

A Learning Agent Shell

for Building Knowledge-Based Agents

Gheorghe Tecuci, Mihai Boicu, Dorin Marcu, Bogdan Stanescu, Cristina Boicu, Marcel Barbulescu

Learning Agents Center, Department of Computer Science MSN 4A5, George Mason University, Fairfax, VA 22030, USA Tel: 1 703 993 {1722, 1591, 1535, 1535, 4669, 1535}

{tecuci, mboicu, dmarcu, bstanesc, ccascava, mbarbule}@gmu.edu

**ABSTRACT**

This paper presents Disciple-RKF, a learning agent shell that can be used by subject matter experts, with limited assistance from knowledge engineers, to develop knowledge-based agents incorporating their expertise.

# Keywords

Knowledge acquisition, learning agent shell, tools for subject matter experts, knowledge engineering, ontology, rules.

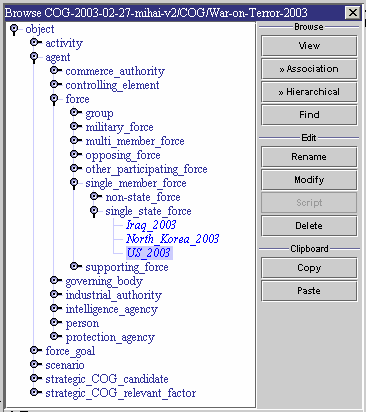
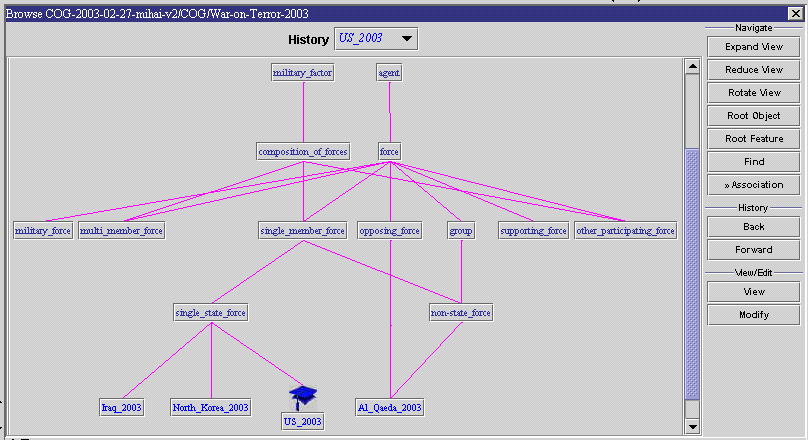
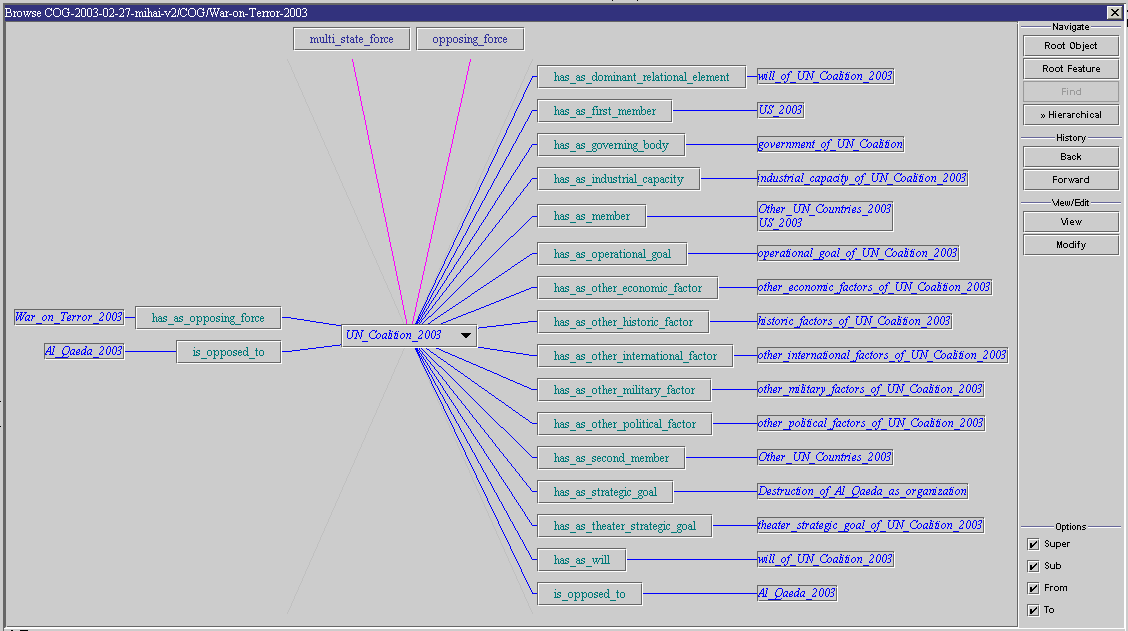
# DISCIPLE-RKF LEARNING AGENT SHELL

Disciple-RKF is a learning agent shell that represents the most

recent implementation of Disciple, an evolving theory and methodology for the development of knowledge bases and agents, by subject matter experts, with limited assistance from knowledge engineers [1, 2, 3]. A main goal of the Disciple approach is to overcome the knowledge acquisition bottleneck in the development of knowledge-based systems.

The knowledge-based agents developed with Disciple-RKF use task-reduction as the main problem solving paradigm. In this paradigm, a problem solving task is successively reduced to simpler tasks, the solutions of the simplest tasks

are found, and these solutions are successively composed into the solution of the initial task. The knowledge base of an agent is structured into an object ontology that represents the objects from an application domain, and a set of task reduction and solution composition rules expressed with these objects. The Disciple-RKF shell is as a general problem solving and learning agent with no specific knowledge in its knowledge base. To develop an agent for a specific application domain, one needs to develop its knowledge base by using the various modules of Disciple-RKF. In essence, this process consists of developing the ontology for the specific application domain and of teaching the agent how to perform various tasks, in a way that resembles how one would teach a human apprentice. In the last four years, successive versions of Disciple-RKF have been used to develop intelligent agents for center of gravity analysis that are used in several courses at the US Army War College.



*European Conference on Knowledge Engineering and Knowledge Management (EKAW’04)*, October 5–8, 2004, Northamptonshire, UK. Copyright resides with the author(s) of this manuscript.

The next section presents an overview of the Disciple-RKF agent development methodology.

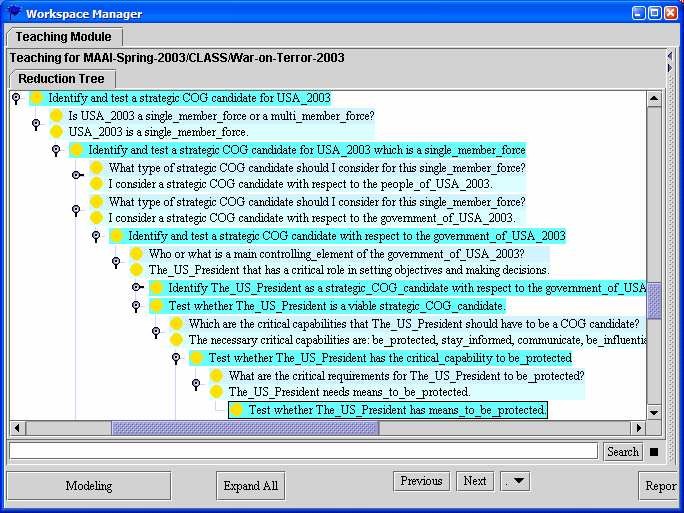
# AGENT DEVELOPMENT METHODOLOGY

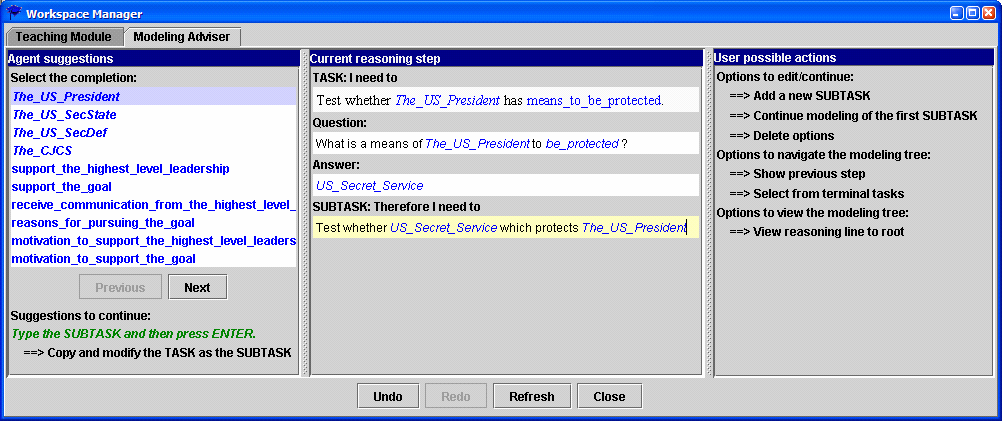
The Disciple approach covers all the phases of agent

development and use. First, a knowledge engineer works with a subject matter expert to develop an ontology for the application domain. They use the ontology import module (to extract relevant ontology elements from existing knowledge repositories) as well as the various ontology editors and browsers of Disciple-RKF. Fig. 1 shows the interfaces of three of Disciple’s ontology browsers: at top is the association browser (which displays an object and its relationships with other objects), at bottom left is the tree browser (which displays the hierarchical relationships between the objects in a tree structure), and at bottom right is the graphical browser (which displays the hierarchical relationships between the objects in a graph structure). The result of this knowledge base development phase is an object ontology which is complete enough to be used as a generalization hierarchy for learning, allowing the expert to train the Disciple agent how to solve problems, with limited assistance from a knowledge engineer.

**Figure 1. Three ontology browsers of Disciple-RKF**

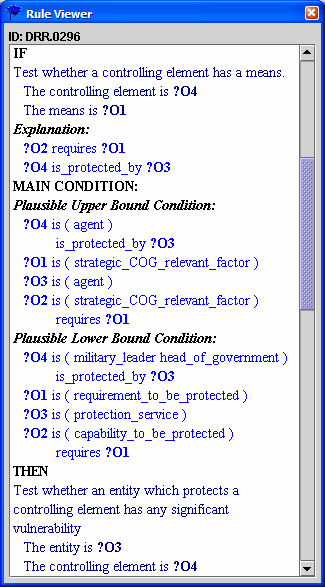
The expert formulates a specific problem solving task and shows the agent the corresponding problem solving steps, helping the agent to understand them. Each problem solving step indicated by the expert consists of a task to be reduced, a question related to that task, the answer to the question, and one or several subtasks or solutions that reduce the task. The top





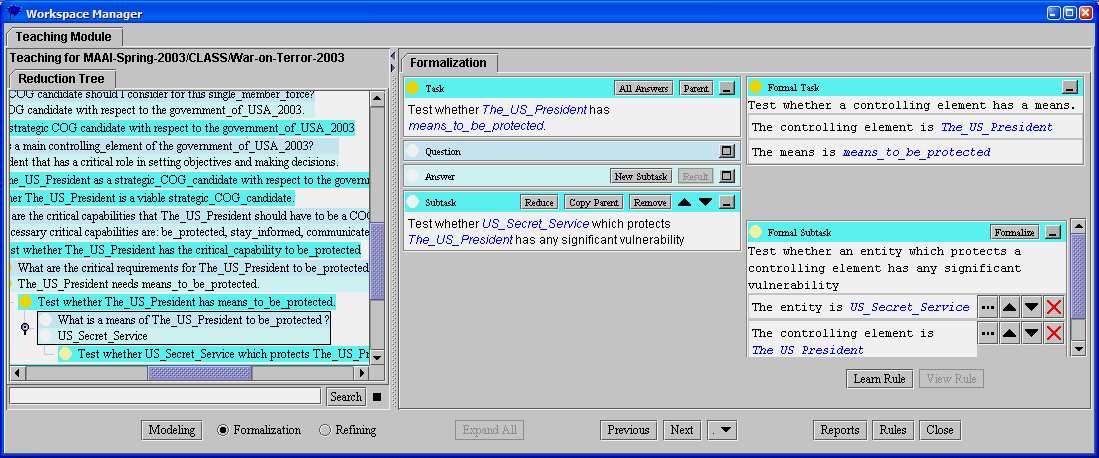
## Figure 2. Reasoning tree and modeling assistant interfaces

part of Fig. 2 shows a fragment of a reasoning tree consisting of a sequence of task reduction steps. The bottom part shows the interface of the Modeling Assistant that helps the expert to express how to reduce the task from the bottom of the reasoning tree. This assistant may suggest the question to be asked, the answer to the question, and the knowledge base elements the expert may be referring to, since the expert’s input is in natural language. Each problem solving step indicated by the expert is an example from which the agent learns a general problem solving rule. First, the expert and Disciple collaborate in formalizing the tasks from the expert’s examples, and Disciple learns general task patterns. Then the expert helps Disciple to find an explanation of why the task reduction step is correct and Disciple learns a general task reduction rule. The top part of Fig. 3 shows the interface of the task formalization module, with the tasks in natural language in the middle and their formalizations in the right. The bottom part of Fig. 3 shows the explanation generation and selection interface. The task reduction rule learned from the example specified in the modeling assistant (see bottom of Fig. 2) is shown in the rule viewer from Fig. 4.

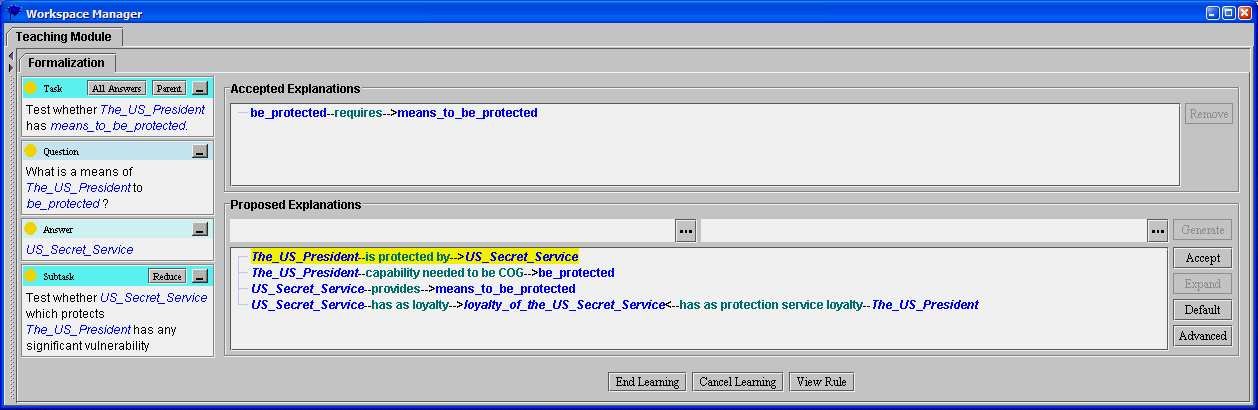
As Disciple learns new rules from the expert, the interaction between the expert and Disciple

evolves from a teacher-student interaction, toward an inter- action where both collaborate in solving a problem. During this mixed-initiative problem solv- ing phase, Disciple learns not only from the contributions of the expert, but also from its own successful or unsuccessful prob- lem solving attempts, which lead to the refinement of the learned rules. At the same time, Disciple extends the object ontology with new objects and features.

Copies of Disciple agents may also be trained in parallel by

different experts. In this case the individual knowledge bases

## Figure 4. Learned rule



**Figure 3. Task formalization and explanation**

have to be merged into an integrated knowledge base, as discussed in [2]. The agent’s knowledge base can also be exported to be used in the development of new agents.

# ACKNOWLEDGEMENTS

This work was supported by the DARPA, AFRL, AFMC, USAF agreement F30602-00-2-0546, by the AFOSR grant F49620-00-1-0072 and by the US Army War College.

# REFERENCES

1. Tecuci, G.:*Building Intelligent Agents: An Apprentice- ship Multistrategy Learning Theory, Methodology, Tool and Case Studies.* Academic Press, London, 1998.
2. Tecuci, G., Boicu, M., Marcu, D., Stanescu, B., Boicu, C., Barbulescu, M., Parallel Knowledge Base Develop- ment by Subject Matter Experts, in *Proc. 14th Inter- national Conference on Knowledge Engineering and Knowledge Management,* EKAW 2004, 5-8th Oct 04 - Northamptonshire, UK, Springer-Verlag, 2004.
3. Disciple-RKF demo at <http://lac.gmu.edu/RKF/cog>

/DISCIPLE-COG%20DEMO.pdf

2