In Proceedings of the Fourteenth Annual Conference on Innovative Applications of Artificial Intelligence, IAAI-2002, Edmonton, Alberta, Canada, July 2002, AAAI Press/The MIT Press, 2002.

Development and Deployment of a Disciple Agent for Center of Gravity Analysis

Gheorghe Tecuci[1,2], Mihai Boicu[1], Dorin Marcu[1], Bogdan Stanescu[1], Cristina Boicu[1], Jerry Comello[2], Antonio Lopez[2], James Donlon[1,2], William Cleckner[2]

[1] Learning Agents Laboratory, Computer Science Department, MS 4A5, George Mason University, 4400 University Dr, Fairfax, VA 22030
{tecuci, mboicu, dmarcu, bstanesc, ccascava}@gmu.edu, http://lalab.gmu.edu
[2] Center for Strategic Leadership, US Army War College, 650 Wright Ave, Carlisle Barracks, PA 17013
ComelloJ@awc.carlisle.army.mil, tlopez@xula.edu, james.donlon@us.army.mil, william.cleckner@csl.carlisle.army.mil

Abstract

This paper presents new significant advances in the Disciple approach for building knowledge-based systems by subject matter experts. It describes the innovative application of this approach to the development of an agent for the analysis of strategic centers of gravity in military conflicts. This application has been deployed in several courses at the US Army War College, and its use has been evaluated. The presented results are those of a multi-faceted research and development effort that synergistically integrates research in Artificial Intelligence, Center of Gravity analysis, and practical deployment of an agent into Education.

1 Introduction

The Learning Agents Laboratory of George Mason University is developing a theory, methodology and a family of tools, called the Disciple approach, for building instructable knowledge-based systems or agents (Tecuci 1998). This effort directly addresses the knowledge acquisition bottleneck which we consider to be one of the major barriers in the development and maintenance of Artificial Intelligence applications. The Disciple approach relies on a powerful learning agent shell that can be trained to solve problems by a subject matter expert, requiring only limited assistance from a knowledge engineer. As an expert system shell (Clancey 1984), the Disciple learning agent shell includes a general problem solving engine that can be reused for multiple applications. In addition, it includes a multistrategy learning engine for building the knowledge base through a mixed-initiative interaction with the subject matter expert.

As the Disciple approach evolved, we have developed a series of increasingly advanced learning agent shells forming the Disciple family. The most recent family member, Disciple-RKF, represents a significant advancement over its predecessors: Disciple-WA (Tecuci et al. 1999) and Disciple-COA (Tecuci et al. 2001). All these three systems were developed as part of the "High Performance Knowledge Bases" and "Rapid Knowledge Formation" programs supported by DARPA and AFOSR (Burke 1999). Both programs emphasized the use of innovative challenge problems to focus and evaluate the research and development efforts.

Disciple-RKF is the result of a multi-objective collaboration between the Learning Agent Laboratory of George Mason University and the Center for Strategic Leadership of the US Army War College that synergistically integrates research in Artificial Intelligence (AI), with research in military Center of Gravity (COG) analysis (Clausewitz 1832), and the practical use of agents in education. The AI research objective is to develop the technology that will enable subject matter experts who do not have computer science or knowledge engineering experience to develop instructable agents that incorporate their problem solving expertise. The COG research objective is to clarify and formalize the process of the identification of the centers of gravity for enemy and friendly forces at the strategic and operational levels of war, and to enable the development of an intelligent assistant for solving this complex problem. Finally, the educational objective is to enhance the educational process of senior military officers through the use of intelligent agent technology. Each of these three objectives is recognized as important and difficult in its own right. Our experience with addressing them together in a synergistic manner has resulted in faster progress in each of them. Moreover, it offers a new perspective on how to combine research in AI, with research in a specialized domain, and with the development and deployment of prototype systems in education and practice.

The paper presents the current status of this research and development effort. The next section presents the COG challenge problem. This is followed by an end-user perspective on the developed Disciple-RKF/COG agent. Section 4 presents an overview of the Disciple-RKF shell and its use to build the Disciple-RKF/COG agent, emphasizing its new capabilities with respect to the previous Disciple shells. Then section 5 discusses the deployment and evaluation of Disciple in two courses at the US Army War College, "Case Studies in Center of Gravity

In Proceedings of the Fourteenth Annual Conference on Innovative Applications of Artificial Intelligence, IAAI-2002, Edmonton, Alberta, Canada, July 2002, AAAI Press/The MIT Press, 2002.

Analysis," and "Military Applications of Artificial Intelligence." The paper concludes with a summary of the synergistic aspects of this collaborative work.

2 The Center of Gravity Problem

The military literature distinguishes between three levels of conflicts: a strategic level focusing on winning wars, an operational level focusing on winning campaigns, and a tactical level focusing on winning battles. One of the most difficult problems that senior military leaders face at the strategic level is the determination and analysis of the centers of gravity for friendly and opposing forces. Originally introduced by Clausewitz in his classical work "On War" (1832), the center of gravity is now understood as representing "those characteristics, capabilities, or localities from which a military force derives its freedom of action, physical strength, or will to fight" (Department of the Army 2001). A combatant should eliminate or influence the enemy's strategic center of gravity, while adequately protecting its own (Giles and Galvin 1996).

Correctly identifying the centers of gravity of the opposing forces is of highest importance in any conflict. Therefore, in the education of strategic leaders at all the US senior military service colleges, there is a great emphasis on the center of gravity analysis (Strange 1996). COG determination requires a wide range of background knowledge, not only from the military domain, but also from the economic, geographic, political, demographic, historic, international, and other domains. In addition, the situation, the adversaries involved, their goals, and their capabilities can vary in important ways from one scenario to another. When performing this analysis, experts rely on their own professional experience and intuitions, without following a rigorous approach. Recognizing these difficulties, the Center for Strategic Leadership of the US Army War College started in 1993 an effort to elicit and formalize the knowledge of a number of experts in center of gravity. This research resulted in a COG monograph (Giles and Galvin 1996), which provided a basis for the application of Disciple to this high value application domain, and to the development of the Disciple-RKF/COG instructable agent presented in the next section.

3 A Disciple Agent for COG Analysis

Disciple-RKF/COG is an agent used in the US Army War College course titled "Case Studies in Center of Gravity Analysis." In this course Disciple-RKF/COG supports the students to develop a center of gravity analysis report for a war scenario.

First, a personal copy of Disciple-RKF/COG guides the student to identify, study and describe the aspects of a scenario (such as the 1945 US invasion of the island of Okinawa) that are relevant for COG analysis. The student-agent interaction takes place as illustrated in Figure 1. The left part of the window is a table of contents, whose elements indicate various aspects of the scenario. When the student selects one such aspect, Disciple asks specific questions intended to acquire from the student a description

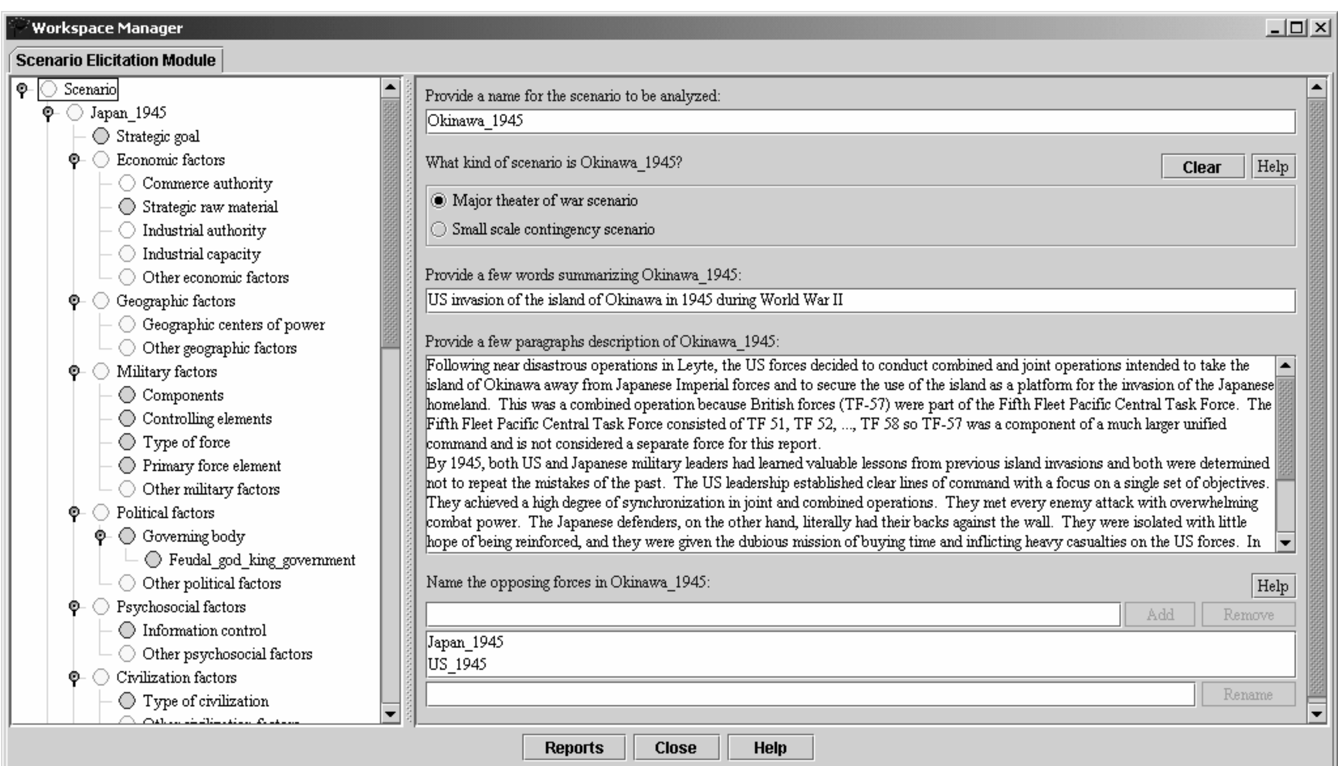

Figure 1: Scenario specification interface

In Proceedings of the Fourteenth Annual Conference on Innovative Applications of Artificial Intelligence, IAAI-2002, Edmonton, Alberta, Canada, July 2002, AAAI Press/The MIT Press, 2002.

of that aspect, or to update a previously specified description. All the answers are in natural language.

Taking the Okinawa_1945 scenario as our example, Disciple starts by asking for a name and a description of the scenario, and then asks for the opposing forces. Once the student indicates Japan_1945 and US_1945 as opposing forces, Disciple includes them in the table of contents, together with their characteristics that the student needs to specify (see the left hand side of Figure 1). Then, the student may click on any of these aspects (e.g. “Industrial capacity” under “Economic factors” of Japan_1945) and the agent guides the student in specifying it. The student’s specification may prompt additional questions from Disciple, and a further expansion of the table of contents. An orange, yellow, or white circle marks each title in the table of contents, indicating respectively that all, some, or none of the corresponding questions of Disciple have been answered. The student is not required to answer all the questions and Disciple can be asked, at any time, to identify and test the strategic center of gravity candidates for the current specification of the scenario.

The right hand side of Figure 2 shows some of the solutions generated by Disciple for the Okinawa_1945 scenario. Each solution identifies an entity as a strategic COG candidate and then indicates whether or not it can be eliminated. In the case of Japan_1945, some of the identified strategic center of gravity candidates are Emperor Hirohito, Japanese Imperial General Staff, the industrial capacity of Japan, and the military of Japan. When a solution is selected in the right hand side of the problem solving interface, its justification, at various levels of abstractions, is displayed in the left hand side.

The left-hand side of Figure 2 shows the detailed justification for the identification and testing of Emperor Hirohito as a strategic COG candidate. Disciple uses the task reduction paradigm to perform the top level problem solving task: “Identify and test a strategic COG candidate for the Okinawa_1945 scenario.” To perform this task, Disciple asks itself a series of questions. The answer of each question allows Disciple to reduce the current task to a simpler one, until Disciple has enough information to first identify a strategic COG candidate, and then to determine whether it should be eliminated or not.

Emperor Hirohito is identified as a strategic COG candidate for Japan_1945 in the Okinawa_1945 scenario because, as the feudal god-king of Japan, he is its main controlling element. After being identified as a candidate, Emperor Hirohito is analyzed based on various elimination tests, but he passes all of them. Because the people of Japan see him as divine, and his will is actually their will, Emperor Hirohito could impose them to accept the unconditional surrender of Japan, which is the main strategic goal of the US. As god-king of Japan and commander in chief of the military, he can also impose his will on the military of Japan. Also, as head of the government, he can impose the government of Japan to accept unconditional surrender. Being able to impose his will on the Clausewitz’s trinity of power (people, military and government), Emperor Hirohito is very likely to be the strategic center of gravity of Japan in 1945.

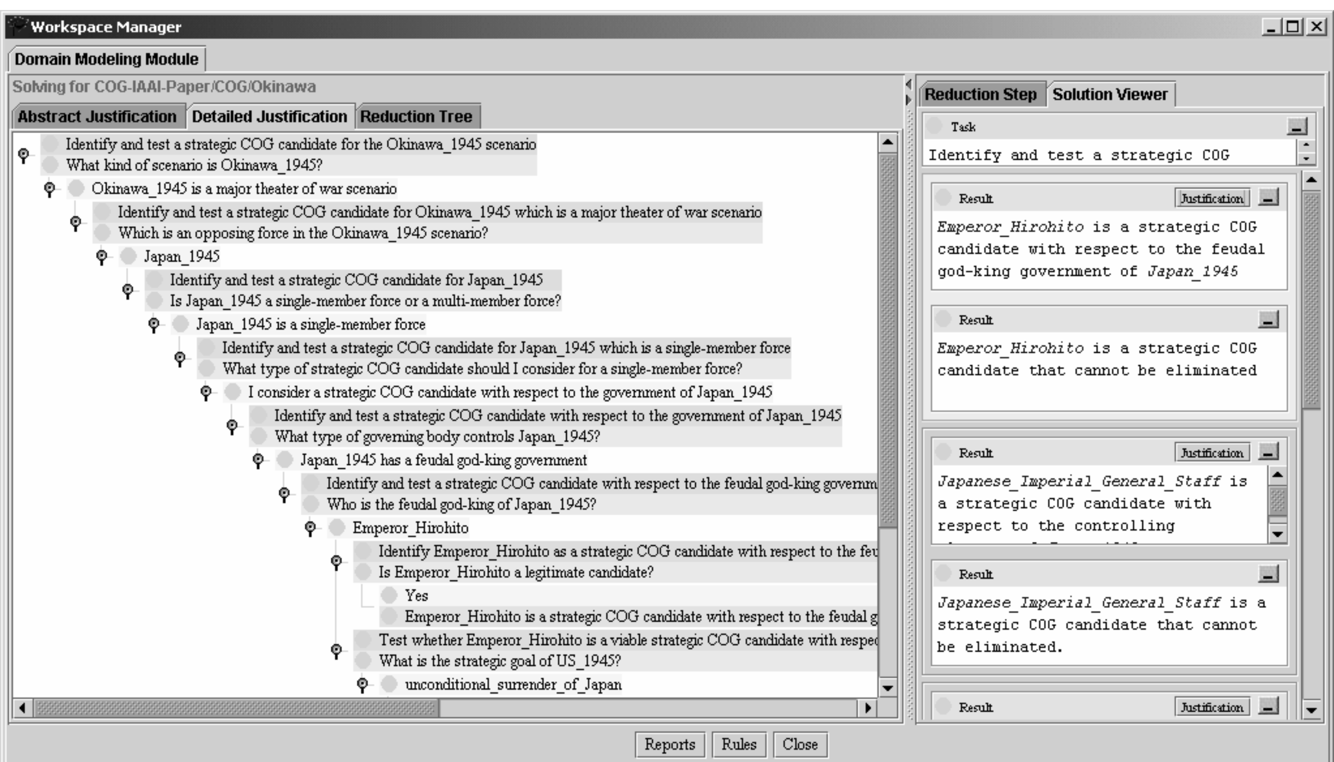

Figure 2: The problem solving interface of Disciple-RKF/COG

In Proceedings of the Fourteenth Annual Conference on Innovative Applications of Artificial Intelligence, IAAI-2002, Edmonton, Alberta, Canada, July 2002, AAAI Press/The MIT Press, 2002.

As another example, consider the industrial capacity of Japan_1945, which is another source of strength, power and resistance because it produces the war materiel and transports of Japan. Disciple, however, eliminates this strategic COG candidate, because the military and the people of Japan_1945 are determined to fight to death and not surrender even with diminished war materiel and transports.

In the example scenario portrayed here, Disciple eliminates all but two candidates for Japan -- Emperor Hirohito and the Japanese Imperial General Staff -- and suggests that the student should select one of them as the strategic Center of Gravity of Japan in 1945. It is important to point out that this example is only a representative approach to the analysis of Japan's center of gravity for the Okinawa campaign. We recognize that subject matter experts often differ in their judgments as to the identification and analysis of center of gravity candidates for any particular scenario. The important point for agent development is that the Disciple agent can accommodate the preferences of the expert who teaches it.

To our knowledge, this is the first time that an intelligent agent for the strategic COG identification and testing has been developed. More details about its specific use in the COG and MAAI classes are presented in section 4.

4 Agent Development with Disciple-RKF

The Disciple-RKF/COG agent presented in the previous section was developed by using the Disciple-RKF learning agent shell, as will be described in this section. Disciple-RKF consists of an integrated set of knowledge acquisition, learning and problem solving modules for a generic knowledge base having two main components: an object ontology that defines the concepts from a specific application domain, and a set of task reduction rules expressed with these concepts. Disciple-RKF represents a significant evolution compared to the previous Disciple shells. It implements more powerful knowledge representation and reasoning mechanisms, and has an improved interface that facilitates mixed-initiative reasoning. Even more significantly, Disciple-RKF incorporates new modules that allow a subject matter expert to perform additional knowledge engineering tasks, such as scenario specification, modeling of his problem solving process, and task formalization.

In general, the process of developing a specific knowledge-based agent with Disciple-RKF consists of two major stages: 1) the development of the object ontology by the knowledge engineer and the subject matter expert, and 2) the training of Disciple by the subject matter expert.

In the first development stage, a knowledge engineer works with a subject matter expert to specify the type of problems to be solved by the Disciple agent, to clarify how these problems could be solved using Disciple's task reduction paradigm, and to develop an object ontology.

A fragment of the object ontology developed for the COG domain is shown in Figure 3. The object ontology

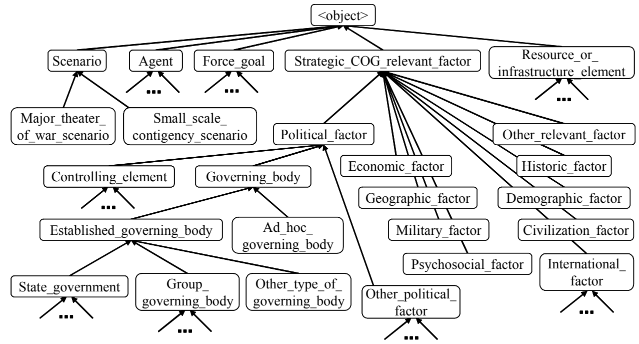

Figure 3: A fragment of the COG object ontology

consists of hierarchical descriptions of objects and features, represented as frames, according to the knowledge model of the Open Knowledge Base Connectivity (OKBC) protocol (Chaudhri et al. 1998). Disciple-RKF includes several types of ontology browsers and editors that facilitate the ontology development process. The careful design and development of the object ontology is of utmost importance because it is used by Disciple as its generalization hierarchy for learning.

A new capability of Disciple-RKF is that ontology development includes the definition of elicitation scripts for objects and features. These scripts guide the expert to define the instances that occur in a scenario (such as Okinawa_1945 or Emperor Hirohito, as illustrated in Section 3). Figure 4 shows the elicitation scripts associated with the "Scenario" object. The top script specifies the question to be asked by Disciple to elicit the name of the scenario, how the user's answer should be used to update the ontology, what other scripts should be called after updating the ontology, and even the appearance of the

Scenario **elicitation_scripts**

Script: Elicit instances of Scenario
Control:
Question: "Provide a name for the scenario to be analyzed:"
Answer variable: <scenario-name>
Default value: new-scenario
Control type: single-line
Ontology action:
<scenario-name> **instance-of** Scenario
Script calls:
Elicit the superconcepts of the instance <scenario-name> of Scenario
Elicit the features of the instance <scenario-name> of Scenario

Script: Elicit the superconcepts of the instance <scenario-name> of Scenario
Control:
Question: "What kind of scenario is" <scenario-name> "?"
Answer variable: <scenario-type>
Default value: Major_theater_of_war_scenario
Control type: single-selection-list
Possible values: the elementary subconcepts of Scenario
Ontology action:
<scenario-name> **instance-of** <scenario-type>

Script: Elicit the features of the instance <scenario-name> of Scenario
Script calls:
Elicit the feature **brief_description** for <scenario-name>
Elicit the feature **description** for <scenario-name>
Elicit the feature **has_as_opposing_force** for <scenario-name>

Figure 4: Sample elicitation scripts

In Proceedings of the Fourteenth Annual Conference on Innovative Applications of Artificial Intelligence, IAAI-2002, Edmonton, Alberta, Canada, July 2002, AAAI Press/The MIT Press, 2002.

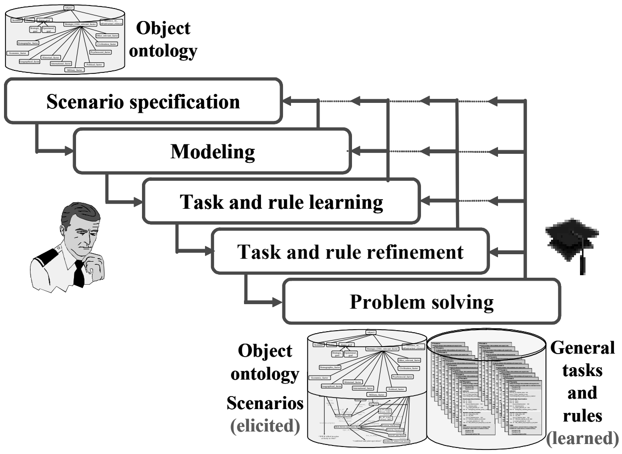

Figure 5: The main phases of the agent training process

interface. The use of the elicitation scripts allows a knowledge engineer to rapidly build customized interfaces for Disciple agents, such as the one illustrated in Figure 1, thus effectively transforming this software development task into a knowledge engineering one.

The result of the first development stage is a customized Disciple agent. This agent is trained to solve problems by a subject matter expert, with very limited assistance from a knowledge engineer, in the second major stage of agent development. Figure 5 shows the main phases of the agent training process, which starts with a knowledge base that contains only a general object ontology (but no instances, no problem solving tasks, and no task reduction rules), and ends with a knowledge base that incorporates expert problem solving knowledge.

During the ***Scenario specification*** phase, the Scenario Specification module (which is a new module of Disciple-RKF) guides the expert in describing the objects that define a specific strategic scenario (e.g. the US invasion of the island of Okinawa in 1945). The expert does not work directly with the object ontology in order to specify the scenario. Instead, the expert-agent interaction takes place as presented in section 3 and illustrated in Figure 1, this all being directed by the execution of the elicitation scripts. Experimental results show that the experts can easily perform this task.

After the expert has specified the Okinawa_1945 scenario, he can start the ***Modeling*** of his COG reasoning for this particular scenario. The expert uses the Modeling module (which is another new module of Disciple-RKF) to express his reasoning in English as a sequence of task reduction steps like the ones illustrated in the left hand side of Figure 2. An example of one task reduction step, defined during modeling, is illustrated in the left hand side of Figure 6. The top task is the current task that needs to be reduced. The expert has to define a question that is relevant to the reduction of this task, then answer the question, and reduce the top task to a simpler one that incorporates the information from the answer. Experimental results show that this is the most challenging agent training activity for the expert.

In the ***Task and rule learning*** phase, Disciple learns general tasks and general rules from the task reduction steps defined in the modeling phase. For instance, consider the reduction step from the left hand side of Figure 6, consisting of a task, a question, an answer, and a subtask. Because all these expressions are in natural language, the expert and the agent collaborate to translate them into the formal logical expressions on the right hand side of Figure 6. First the natural language expression of each task is structured into an abstract phrase called the task name, which does not contain any instance, and several specific phrases representing the task's features. The formalization is proposed by the agent and may be modified by the expert. Next the expert and the agent collaborate to also formalize the question and the answer from the left hand side of Figure 6 into the explanation on the right hand side of Figure 6. This explanation represents the best approximation of the meaning of the question-answer pair that can be formed with the elements of the object ontology. In essence, the agent will use analogical

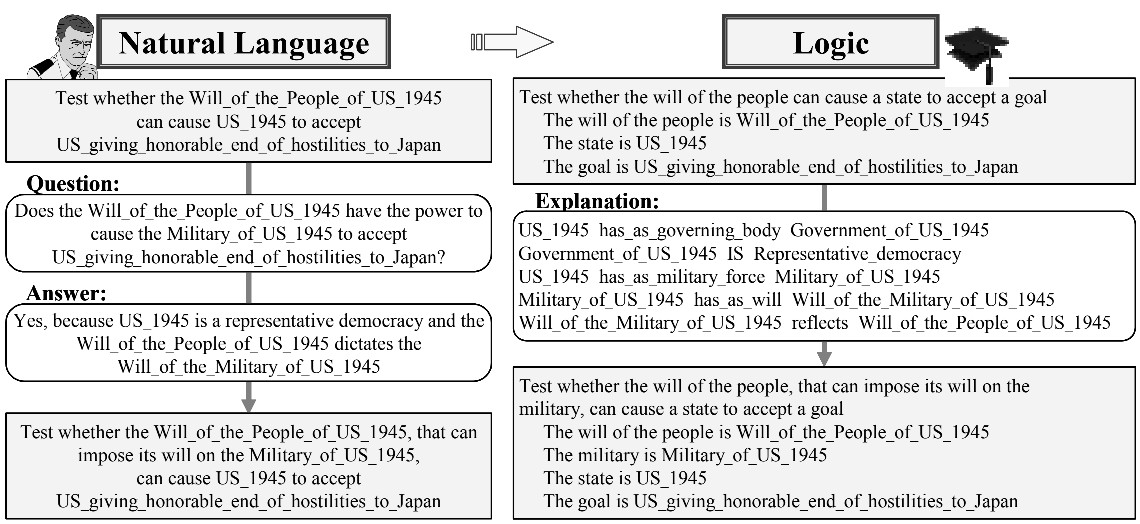

Figure 6: Mixed-initiative language to logic translation

In Proceedings of the Fourteenth Annual Conference on Innovative Applications of Artificial Intelligence, IAAI-2002, Edmonton, Alberta, Canada, July 2002, AAAI Press/The MIT Press, 2002.

Test whether the ?O1 can cause ?O2 to accept ?O3

Test whether the will of the people can cause a state to accept a goal
The will of the people is ?O1
The state is ?O2
The goal is ?O3

Plausible Upper Bound Condition
?O1 is Strategic_cog_relevant_factor
?O2 is Agent
is Strategic_cog_relevant_factor
?O3 is Force_goal

Plausible Lower Bound Condition
?O1 is Will_of_the_People_of_US_1945
?O2 is US_1945
?O3 is US_giving_honorable_end_of_hostilities_to_Japan

Figure 7: Task learned from the example in Figure 6

reasoning and guidance from the expert to propose a set of plausible explanation pieces from which the expert will select the most appropriate ones (Tecuci et al. 2001).

Based on the formalizations from Figure 6 and the object ontology from Figure 3, the Disciple agent learns the general task shown in Figure 7 and the general rule shown in Figure 8. Both the learned task and the learned rule have an informal structure (that preserves the natural language of the expert and is used in agent-user communication), and a formal structure (that is used in the internal formal reasoning of the agent).

Initially, when the agent has no rules and no tasks, the expert teaches Disciple how to solve problems and Disciple generates partially learned tasks and rules, as indicated above. As Disciple learns from the expert, the interaction between the expert and Disciple evolves from a teacher-student interaction, toward an interaction where both collaborate in solving a problem. During this mixed-initiative ***Problem Solving*** phase, Disciple learns not only from the contributions of the expert, but also from its own successful or unsuccessful problem solving attempts.

The learned formal rule in Figure 8 includes two applicability conditions, a plausible upper bound (PUB) condition, and a plausible lower bound (PLB) condition. The PUB condition allows the rule to be applicable in many analogous situations, but the result may not be correct. The PLB condition allows the rule to be applicable only in the situation from which the rule was learned. The agent will apply this rule to solve new problems and the feedback received from the expert will be used to further refine the rule. In essence, the two conditions will converge toward one another (usually through the specialization of the PUB condition and the generalization of the PLB condition), both approaching the exact applicability condition of the rule. ***Rule refinement*** could lead to a complex task reduction rule, with additional Except-When conditions which should not be satisfied in order for the rule to be applicable. Tasks are refined in a similar way.

It is important to stress that the expert does not deal directly with the learned tasks and rules, but only with their examples used in problem solving. Therefore, the complex knowledge engineering operations of defining and debugging problem solving rules are replaced in the Disciple approach with the much simpler operations of defining and critiquing specific examples.

After the Disciple agent has been trained, it can be used in the autonomous problem solving mode, to identify and test the strategic COG candidates for a new scenario, as was illustrated in Section 3.

5 Deployment and Evaluation of Disciple-RKF/COG

The US Army War College regularly offers the courses "Case Studies in Center of Gravity Analysis" and "Military Applications of Artificial intelligence." In the first course (the COG course), the students use Disciple-RKF/COG as an intelligent assistant that supports them to develop a center of gravity analysis report for a war scenario, as described in section 3. In the second course (the MAAI course), the students act as subject matter experts that teach personal Disciple-RKF agents their own reasoning in

IF: Test whether the ?O1 can cause ?O2 to accept ?O3
Question: Does the ?O1 have the power to cause the ?O4 to accept ?O3?
Answer: Yes, because ?O2 is a representative democracy and the ?O1 dictates the ?O5
THEN: Test whether the ?O1, that can impose its will on the ?O4, can cause ?O2 to accept ?O3

IF: Test whether the will of the people can cause a state to accept a goal
The will of the people is ?O1
The state is ?O2
The goal is ?O3

Explanation
?O2 has_as_governing_body ?O6
?O6 IS Representative_democracy
?O2 has_as_military_force ?O4 has_as_will ?O5 reflects ?O1

Plausible Upper Bound Condition
?O1 is Will_of_agent
?O2 is Force
has_as_military_force ?O4
has_as_governing_body ?O6
?O3 is Force_goal
?O4 is Military_force
has_as_will ?O5
?O5 is Will_of_agent
reflects ?O1
?O6 is Representative_democracy

Plausible Lower Bound Condition
?O1 is Will_of_the_People_of_US_1945
?O2 is US_1945
has_as_military_force ?O4
has_as_governing_body ?O6
?O3 is US_giving_honorable_end_of_hostilities_to_Japan
?O4 is Military_of_US_1945
has_as_will ?O5
?O5 is Will_of_the_Military_of_US_1945
reflects ?O1
?O6 is Government_of_US_1945

THEN: Test whether the will of the people, that can impose its will on the military, can cause a state to accept a goal
The will of the people is ?O1
The military is ?O4
The state is ?O2
The goal is ?O3

Figure 8: Rule learned from the example in Figure 6

Center of Gravity analysis, as described in section 4.

As briefly illustrated in section 3, Disciple-RKF/COG guides the student to identify, study and describe the relevant aspects of the opposing forces in a particular scenario. Then Disciple identifies and tests the strategic center of gravity candidates and generates a draft analysis report that the student needs to finalize. The student must examine Disciple's reasoning, correct or complete it, or even reject it and provide an alternative line of reasoning. This is productive for several reasons. First the given agent generates its proposed solutions by applying general reasoning rules and heuristics learned previously from the course's instructor, to a new scenario described by the student. Secondly, COG analysis is influenced by personal experiences and subjective judgments, and the student (who has unique military experience and biases) may have a different interpretation of certain facts.

This requirement for the critical analysis of the solutions generated by the agent is an important educational component of military commanders that mimics military practice. Commanders have to critically investigate several courses of actions proposed by their staff and to make the final decision on which one to use.

During the 2001 academic year, Disciple was successfully used in both the Winter and Spring sessions of the COG course. As a result of this initial success, the USAWC decided to continue and expand the integration of Disciple in this course for the next academic year and beyond. At the end of the courses the students completed detailed evaluation forms about Disciple and its modules, addressing a wide range of issues, ranging from judging its usefulness in achieving course's objectives, to judging its methodological approach to problem solving, and to judging the ease of use and other aspects of various modules. For instance, on a 5-point scale, from strongly disagree to strongly agree, 7 out of 13 students agreed and the other 6 s trongly agreed that the Scenario Specification module should be used in future versions of the course. Furthermore, 8 out of 11 students agreed, 1 strongly agreed, and 2 were neutral that subject matter experts who are not computer scientists can learn to express their reasoning process using the task reduction paradigm. As another example, 10 out of 13 students agreed, 1 strongly agreed, 1 disagreed and 1 strongly disagreed that the Scenario Specification tool is easy to use.

Several of the students that took the COG course in the Winter 2001 session, together with additional students, took the "Military Applications of Artificial Intelligence Course" in the Spring 2001 session. In this course the students were given a general overview of Artificial Intelligence, as well as an introduction to Disciple-RKF. These students used Disciple-RKF as subject matter experts. The students were organized in five two-person teams, with each team given the project to train a personal Disciple-RKF agent shell according to its own reasoning in COG identification for its historical scenario. All five teams succeeded in developing working agents, with each team addressing a unique scenario: 1) the capture of the Leyte Island by the US forces in 1944; 2) the Inchon landing during the Korean War in 1950; 3) the Falklands war between Argentina and Britain in 1982; 4) the stabilization mission in the Grenada Island in 1983; and 5) the US invasion of Panama in December 1989.

In the last two 3-hour class sessions, all the five teams participated in a controlled agent development experiment that was videotaped in its entirety. Each team was again provided with a copy of Disciple-RKF that contained the generic object ontology from Figure 3, but no specific instances, no tasks and no rules. They received a 7-page report describing the Okinawa scenario, and were asked to train their Disciple agent to identify center of gravity candidates, based on that scenario. After each significant phase of agent training and knowledge base development (i.e. scenario specification, modeling, rule learning, and rule refinement) a knowledge engineer reviewed their work, and the team then made any necessary corrections under the supervision of the knowledge engineer. Each team succeeded in specifying the scenario and training the agent, in a very short time, as indicated in Figure 9.

The top table in Figure 9 shows the size of the initial object ontology. Each team interacted with Disciple to populate this ontology with different instances and features representing the Okinawa scenario. After that, each team taught its Disciple agent to identify COG candidates for this scenario. The bottom table in Figure 9 indicates both the time spent by each team, and the number of knowledge elements defined during this time. On average they defined 85.4 instances and 93.8 feature values in 1 hour and 6 minutes. The average number of rules per team was 18.8, and the average time interval was 4 hours and 7 minutes. Although obviously incomplete (both because of the use of a single training scenario, and because of incomplete training for that scenario), the knowledge bases were good enough for identifying correct COG candidates not only for the Okinawa (evaluation) scenario, but also for "new" scenarios whose inputs were taken from the class projects.

At the end of this final experiment, the students completed a detailed questionnaire, containing questions about the main components of Disciple. One of the most significant results was that 7 out of the 10 experts agreed, 1 expert strongly agreed and 2 experts were neutral with respect to the statement: "I think that a subject matter expert can use Disciple to build an agent, with limited assistance from a knowledge engineer." We consider this experiment to be a very significant success. Indeed, to our knowledge, this is the first time that subject matter experts

Initial KB	Generic Object Ontology	
	Concepts	Features
	144	79

Teams	Okinawa Scenario			General Tasks and Rules		
	Instances	Feat Val	Time	Tasks	Rules	Time
Team 1	94	103	1h 21min	18	17	3h 52min
Team 2	78	86	55min	23	22	4h 21min
Team 3	72	79	52min	22	19	4h 35min
Team 4	105	111	1h 23min	18	17	3h 58min
Team 5	78	90	59min	20	19	3h 46min
Average	85.4	93.8	1h 06min	20.2	18.8	4h 07min

Figure 9: Knowledge base development during the final experiment

have trained an agent their own problem solving expertise, with very limited assistance from a knowledge engineer.

The deployment and evaluation of Disciple have also revealed several limitations of this approach and have provided numerous ideas for improvement. For instance, while the subject matter expert has an increased role and independence in agent development, the knowledge engineer still has a critical role to play. He has to assure the development of a fairly complete object ontology. He also has to develop a generic modeling of the problem solving process based on the task reduction paradigm. Even guided by this generic modeling, and using natural language, the subject matter expert has difficulties in expressing his reasoning process. Therefore more work is needed to develop methods for helping the expert in this task.

Several other research groups are addressing the problem of direct knowledge acquisition from subject matter experts, as part of the DARPA's "Rapid Knowledge Formation" program (Burke 1999). These groups, however, are currently emphasizing the acquisition of textbook knowledge, relying primarily on reusing knowledge from existing knowledge repositories. In contrast, the emphasis of our research is on acquiring expert's problem solving knowledge that is not normally represented in textbooks, and relies primarily on teaching and learning.

6 Conclusions

This paper presented the current status of a multi-faceted research and development effort that synergistically integrates research in AI, research in COG analysis, and the practical application to education.

The AI research in knowledge bases and agent development by subject matter experts has benefited from the COG domain that provided a complex challenge problem. Identification and testing of strategic COG candidates exemplifies expert problem solving that relies on a wide range of domain knowledge, a significant part of which is tacit. This research has also benefited from its practical application to education. The COG and MAAI courses allowed us to perform thorough experimentation with real experts, resulting in the validation of our methods and providing ideas for future improvements.

The research in COG analysis has benefited from the AI research in that the agent development has helped clarify and formalize the COG identification and testing process. The COG reasoning models developed were validated in the COG and MAAI classes, and are leading to a significant revision and improvement of the COG monograph of Giles and Galvin (1996).

Finally, the innovative application of the AI and COG research to education through the use of the Disciple agent, has had a significant impact on improving the COG and MAAI courses. Done as a very successful experiment in 2001, it was made a regular part of the syllabi for 2002, to be continued in the following years.

Acknowledgments

This paper is dedicated to Mihai Draganescu who supported the first research projects on the Disciple approach at the Research Institute for Informatics and the Romanian Academy. The research described in this paper was sponsored by DARPA, AFRL, AFMC, USAF, under agreement number F30602-00-2-0546, by AFOSR under grant no. F49620-00-1-0072, and by the US Army War College. Michael Bowman, Catalin Balan, Marcel Barbulescu, Xianjun Hao and other members of the LALAB contributed to the development of Disciple-RKF.

References

Burke, M. 1999. Rapid Knowledge Formation Program Description, http://www.darpa.mil/ito/research/rkf/

Chaudhri, V. K.; Farquhar, A.; Fikes, R.; Park, P. D.; and Rice, J. P. 1998. OKBC. A Programmatic Foundation for Knowledge Base Interoperability. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, 600–607. Menlo Park, California: AAAI Press.

Clancey, W. J. 1984. NEOMYCIN: Reconfiguring a rule-based system with application to teaching. In Clancey W. J. and Shortliffe, E. H. eds. *Readings in Medical Artificial Intelligence,* 361-381. Reading, MA: Addison-Wesley.

Clausewitz, C.V. 1832. *On War,* translated and edited by M. Howard and P. Paret. Princeton, NJ: Princeton University Press, 1976.

Department of the Army 2001. *Field Manual 3-0, Operations*. Washington, D.C.: U.S. Gov. Printing Office.

Giles, P.K., and Galvin, T.P. 1996. *Center of Gravity: Determination, Analysis and Application*. CSL, U.S. Army War College, PA: Carlisle Barracks.

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

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

Tecuci, G.; Boicu, M.; Wright, K.; Lee, S. W.; Marcu, D.; and Bowman, M. 1999. An Integrated Shell and Methodology for Rapid Development of Knowledge-Based Agents. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, 250-257. Menlo Park, California: AAAI Press.

Tecuci G.; Boicu M.; Bowman M.; and Marcu D. with a commentary by Burke M. 2001. An Innovative Application from the DARPA Knowledge Bases Programs: Rapid Development of a High Performance Knowledge Base for Course of Action Critiquing. *AI Magazine* Vol. 22, No. 2: 43-61.