Studies in Informatics and Control, December 2007, Volume 16, Number 4, pp. 459-468.

Mixed-Initiative Assumption-Based Reasoning for Complex Decision-Making

Gheorghe Tecuci, Dorin Marcu, Mihai Boicu, Vu Le

Learning Agents Center, George Mason University, Fairfax, VA 22032, USA
{tecuci, dmarcu, mboicu, vle3}@gmu.edu, http://lac.gmu.edu

Abstract: This paper discusses several critical capabilities of the Disciple-LTA system for complex problem-solving and decision-making, including a transparent and easy to understand reasoning process, a flexible and natural collaboration with the user, and the use of what-if scenarios to cope with incomplete and uncertain information. They allow the user to act as the orchestrator of the reasoning process, guiding the high-level exploration of the decision-making space, while the system implements this guidance by taking into account the user's preferred problem solving strategies, assumptions and biases. These capabilities are discussed in the context of intelligence analysis.

Keywords: decision support systems, problem reduction and solution synthesis, assumption-based reasoning, mixed-initiative interaction, learning, intelligence analysis.

Gheorghe Tecuci is Professor of Computer Science in the Volgenau School of Information Technology and Engineering and Director of the Learning Agents Center at George Mason University. He is also a member of the Romanian Academy, honorary director of its Research Institute for Artificial Intelligence, and former Chair of Artificial Intelligence at the US Army War College.

Dorin Marcu is a Ph.D. candidate in Computer Science and a Graduate Research Assistant in the Learning Agents Center at George Mason University. He holds a B.S. in Computer Science from Polytechnic University of Bucharest, Romania, and has published over 30 conference and journal papers.

Mihai Boicu is Assistant Professor of Information Technology and Associate Director of the Learning Agents Center at George Mason University. He received a License in Informatics from the Bucharest University in 1995, and a Ph.D. in Information Technology from George Mason University in 2003. He has published around 50 papers.

Vu Le is a Research Instructor in the Learning Agents Center and a Ph.D. candidate in Information Technology at George Mason University. He received an M.S. in Computer Science, in 1999, from George Mason University and has published several papers.

1. Introduction

The use of computer systems to aid in problem solving and decision making has long been an area of research and development [1, 2, 3, 4, 5, 6]. A main motivation of this effort is the obvious complementariness between humans and computers with respect to this complex reasoning process. Humans are slow, sloppy, forgetful, implicit, and subjective, but have common sense and intuition, and may find creative solutions in new situations. In contrast, computer systems are fast, rigorous, precise, explicit, and objective, but they lack common sense and the ability to deal with new situations [7]. Moreover, in contrast to a computer system, a human has a very limited attention span and can analyze only a small number of alternatives at a time [8]. Therefore, a main research objective is to create an environment for problem solving and decision making that synergistically integrates the complementary capabilities of humans and computer systems, taking advantage of their relative strengths to compensate for each-others weaknesses. This objective becomes even more important in face of the globalization and the rapid evolution toward the knowledge economies [9, 10] which add additional challenges to decision-makers who need to cope with dynamic and increasingly complex situations, and make good decisions in face of an overwhelming amount of incomplete, uncertain, and mostly irrelevant information.

We think that a good problem-solving and decision-making environment is one where the human acts as the orchestrator of the reasoning process, guiding the high-level exploration, while the computer system implements this guidance by taking into account the human's preferred problem solving strategies, assumptions and biases [11]. In such an environment, the computer system is an extension of the reasoning capabilities of the human, much like a calculator is an extension of the computational capabilities of an accountant. The emphasis is on enhancing human's creativity [12], relying on the human to take the most critical decisions, and only to critique and correct the more routine ones that are proposed by the computer system [11]. To develop such an environment requires an automatic approach to problem solving which is very natural and easy to understand. Moreover, the human and the computer should collaborate in a natural way, similarly to how humans collaborate, as opposed to the usual human-computer interaction which is inflexible and mostly unidirectional. Also, because most of the complex decisions are based on incomplete and uncertain information, the decision-making environment should allow the investigation of what-if scenarios, where the decision-maker can make various assumptions about a situation.

For many years we have investigated an approach to the development of knowledge-based computer assistants that would have the above capabilities. The result is an evolving theory, methodology, and family of software tools, collectively known as the Disciple approach [13, 14, 15]. Several experimental Disciple assistants have been developed to support decision makers in different domains, including course of action critiquing [16], military center of gravity analysis [17], emergency response planning [18] and intelligence analysis [11].

The next section provides a general overview of the Disciple approach and introduces its application to intelligence analysis. Then, Section 3 presents Disciple's capability to investigate what-if scenario through the use of assumptions. Section 4 presents the mixed-initiative interaction framework implemented in Disciple and how it facilitates the personalization of the interaction with the user. Section 5 concludes the paper with some results of an evaluation performed at the US Army War College and a discussion of future research.

2. Disciple Assistants for Problem-Solving and Decision-Making

Disciple denotes an evolving theory, a methodology and a set of software tools for building agents that incorporate the problem solving knowledge of a human expert. The guiding idea behind the Disciple approach is to create a generic agent that can be taught directly by a human expert (who is not a computer scientist or a knowledge engineer) in a way that resembles how the expert would teach a student or an apprentice, while solving problems in collaboration. For example, the expert may select a specific problem and may explain the agent how to solve it, or s/he may critique the agent's attempts to solve it.

Using advanced machine learning methods [15, 17], the agent will learn or refine general reasoning rules that will allow it to solve similar problems.

Disciple employs a general divide-and-conquer approach to problem solving, known as problem-reduction/solution-synthesis [19, 20, 13]. In this approach, illustrated in Figure 1, a complex problem (such as “Assess whether Terrorist Group A has nuclear weapons.”) is successively reduced to simpler and simpler problems, the solutions of the simplest problems are found, and then these solutions are successively composed, from bottom up, until the solution of the initial problem is obtained (for example, “It is likely that Terrorist Group A has nuclear weapons.”). This general problem-reduction/solution-synthesis approach is customized for a given application domain, such as intelligence analysis, as illustrated in the left-hand side of Figure 1. In this domain, the problems consist of testing complex hypotheses based on incomplete and uncertain information contained in the available pieces of evidence (e.g. newspaper articles, web sites, news agency reports, books) which provide some relevant information that may not be entirely believable.

1) A complex hypothesis analysis problem is successively reduced to simpler problems that either have known solutions or can be solved through evidence analysis.
2) Potentially relevant pieces of evidence for the unsolved problems are identified.
3) The pieces of evidence are analyzed to obtain solutions for the unsolved problems.
4) The solutions of the simplest problems are successively combined to obtain the solution of the initial problem.

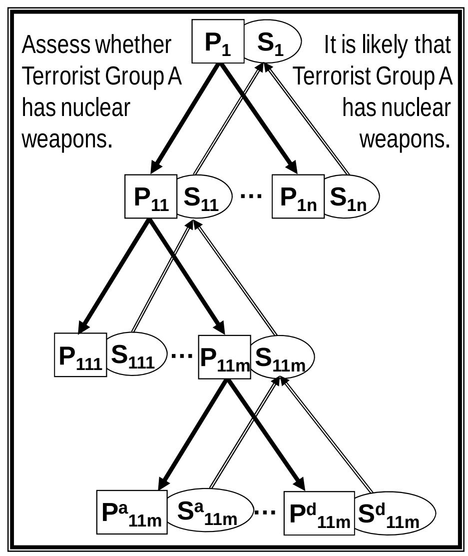

Figure 1: The problem-reduction/solution synthesis paradigm applied to intelligence analysis.

To exhibit this type of problem solving behavior, the knowledge base of a Disciple agent should contain an ontology (which describes the terms from an application domain, such as piece of evidence, tangible evidence, testimonial evidence, credibility, accuracy) and problem solving rules (expressed with the terms from the ontology). One type of rule is the problem reduction rule which expresses how and under what conditions a generic problem can be reduced to simpler generic problems. Another type of rule is the solution synthesis rule which expresses how and under what conditions the solutions of generic sub-problems can be combined into the solution of a generic problem.

During agent training, a subject matter expert (expert analyst, in our example) will collaborate with Disciple to analyze a specific hypothesis by developing a reasoning tree. During this process, Disciple will learn reduction and synthesis rules from the reasoning steps contributed by the expert, and will improve previously learned rules based on the expert’s critique of their use. Disciple includes several assistants that help the expert to teach it, such as the Modeling Assistant (for defining concrete problem solving examples from which Disciple learns general rules [21]), the Explanation Generation Assistant (for explaining the problem solving examples during rule learning and refinement [17]), the Rule Refinement Assistant (for focusing the user on the reasoning steps that need to be critiqued [22]), and the Elicitation Assistant (for eliciting knowledge about the current situation).

After is trained, a Disciple agent can solve problems similarly to how the expert instructed it. For example, Figure 2 shows the reduction tree generated by Disciple-LTA (the Disciple system for intelligence analysis) for the hypothesis analysis problem "Assess whether Terrorist Group A has nuclear weapons." As the expert would, Disciple-LTA asks itself a question on how to reduce this problem (i.e. "What factors should I consider to determine whether Terrorist Group A has nuclear weapons?"). The answer ("Characteristics associated with possession of nuclear weapons and current evidence that it has nuclear weapons.") leads Disciple-LTA to reduce the initial problem to two simpler problems:

> "Assess whether Terrorist Group A has nuclear weapons based on the characteristics associated with the possession of nuclear weapons."
>
> "Assess whether there is current evidence that Terrorist Group A has nuclear weapons."

Each of these two hypothesis analysis problems is reduced in a similar way, guided by questions and answers, as illustrated in the right-hand side of Figure 2. For example, the first problem is reduced to three simpler problems:

> "Assess whether Terrorist Group A has reasons to obtain nuclear weapons."
>
> "Assess whether Terrorist Group A has desire to obtain nuclear weapons."
>
> "Assess whether Terrorist Group A has the ability to obtain nuclear weapons."

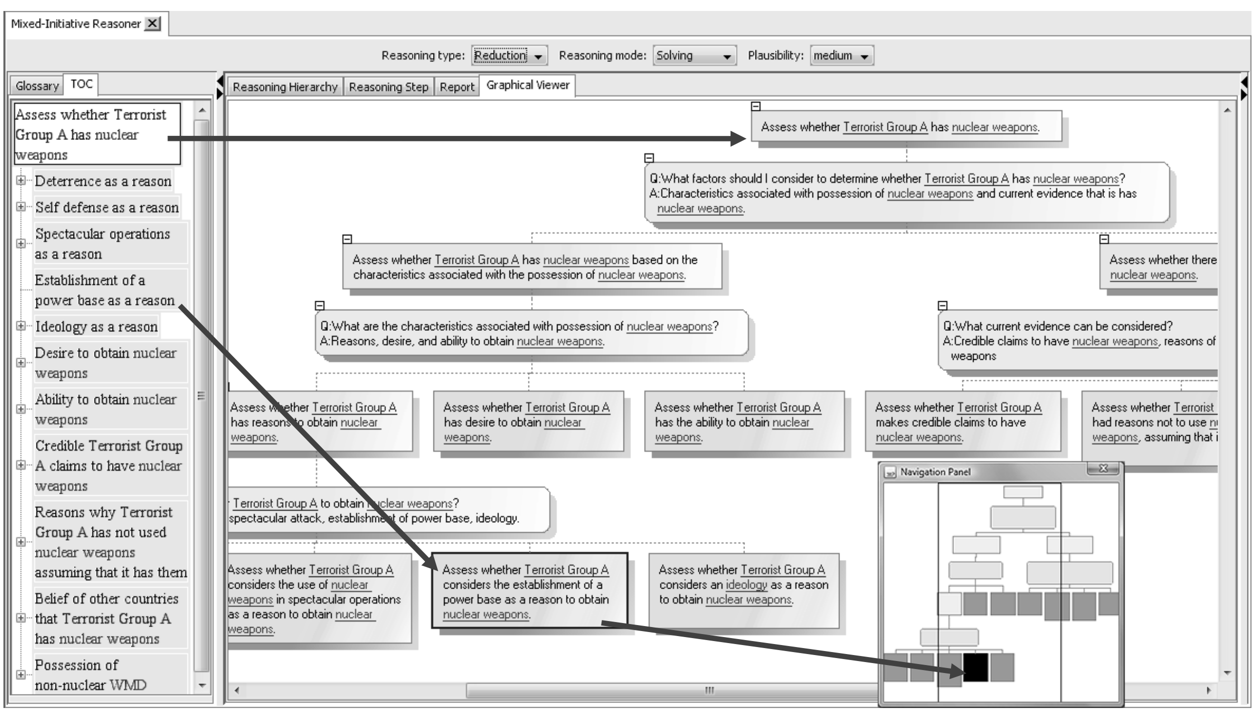

Figure 2: Hypothesis analysis through problem reduction.

The generated reasoning trees are very large for complex problems, with thousands of nodes. In order to facilitate their browsing and understanding by the user, these reasoning trees are abstracted and partitioned in manageable sub-trees, as illustrated in Figure 2. In particular, each of the sub-problems in the left-hand side of Figure 2 (e.g. "Establishment of a power base as a reason") is the abstraction of a sub-problem shown as a leaf in the sub-tree displayed in the right-hand side of Figure 2 (i.e. "Assess whether Terrorist Group A considers establishment of a power base as a reason to obtain nuclear weapons."). Thus, the right hand-side of Figure 2 shows the detailed reasoning for reducing a complex problem to a set of simpler sub-problems (shown as leaves in the sub-tree), while the left-hand side of Figure 2 shows the abstractions of these simpler (leaf) sub-problems. When the user clicks on an abstract sub-problem, Disciple-LTA displays its reduction sub-tree to even simpler leaf sub-problems, and so on. Therefore, left hand side of Figure 2 is an abstraction of the main sub-problems from the reasoning tree, and plays the role of a table of content (TOC) for navigating the entire reasoning tree.

Disciple-LTA successively reduces the initial hypothesis analysis problem to simpler problems until elementary hypothesis analysis problems are reached. For each elementary hypothesis Disciple-LTA identifies relevant pieces of evidence and determines to what extent they favor or disfavor that hypothesis. The analysis of each piece of evidence takes into account its chain of custody, as well as the competence and the credibility of the corresponding primary and intermediary sources [11]. Consider, for example, "Self defense as a reason", the second sub-problem from the left-hand side of Figure 2. This is the abstraction of "Assess whether Terrorist Group A considers self defense as a reason to obtain nuclear weapons" which represents an elementary hypothesis analysis problem. To solve this problem Disciple-LTA looks both for pieces of evidence that favor the hypothesis that "Terrorist Group A considers self defense as a reason to obtain nuclear weapons" and for pieces of evidence that disfavor this hypothesis. A favoring piece of evidence is a published interview with Leader X of Terrorist Group A where he claims that "Terrorist Group A has nuclear weapons and may use them to defend itself." If one would believe this piece of evidence, then the hypothesis would be true. But how believable is it? Leader X's statement was conveyed by a reporter who may have distorted it. Thus, to assess the believability of this piece of evidence, Disciple-LTA would need to assess both the believability of the primary source of information (i.e. Leader X), and the believability of the secondary source (i.e. the reporter). Furthermore, to assess the believability of each of these sources, Disciple-LTA would need to assess their competence and their credibility [23]. To assess the credibility of Leader X, Disciple-LTA would need to assess his veracity (i.e. the degree to which he believes he is telling the truth), his objectivity (i.e. the degree to which his judgment is based on observable phenomena, uninfluenced by emotions or personal prejudices), and his observational sensitivity (i.e. the degree to which his senses give evidence to what he said). All these assessments are based on incomplete, uncertain, and/or contradictory information, and are expressed as symbolic probabilities, such as, remote, unlikely, even chance, likely, almost certain (e.g. "The observational sensitivity of Leader X with respect to the information provided in EVD-NewspaperU-ReporterV-01-01 is almost certain."). Once these lower level assessments are performed, they are combined, from bottom up, to assess the credibility of Leader X, then his believability and the believability of the reporter, then the believability of the entire piece of evidence, then the extent to which this piece of evidence supports the hypothesis that "Terrorist Group A considers self defense as a reason to obtain nuclear weapons", and so on, until Disciple-LTA obtains the solution of the elementary hypothesis analysis problem "Assess whether Terrorist Group A considers self defense as a reason to obtain nuclear weapons" (i.e. "It is an even chance that Terrorist

Group A considers self defense as a reason to obtain nuclear weapons."). The solutions of all such elementary hypothesis analysis problems are then successively combined to produce the solution for the initial problem ("It is almost certain that Terrorist Group A has nuclear weapons."). This is illustrated in Figure 3 where each problem is represented as a node in the tree, and each solution is displayed below its corresponding problem.

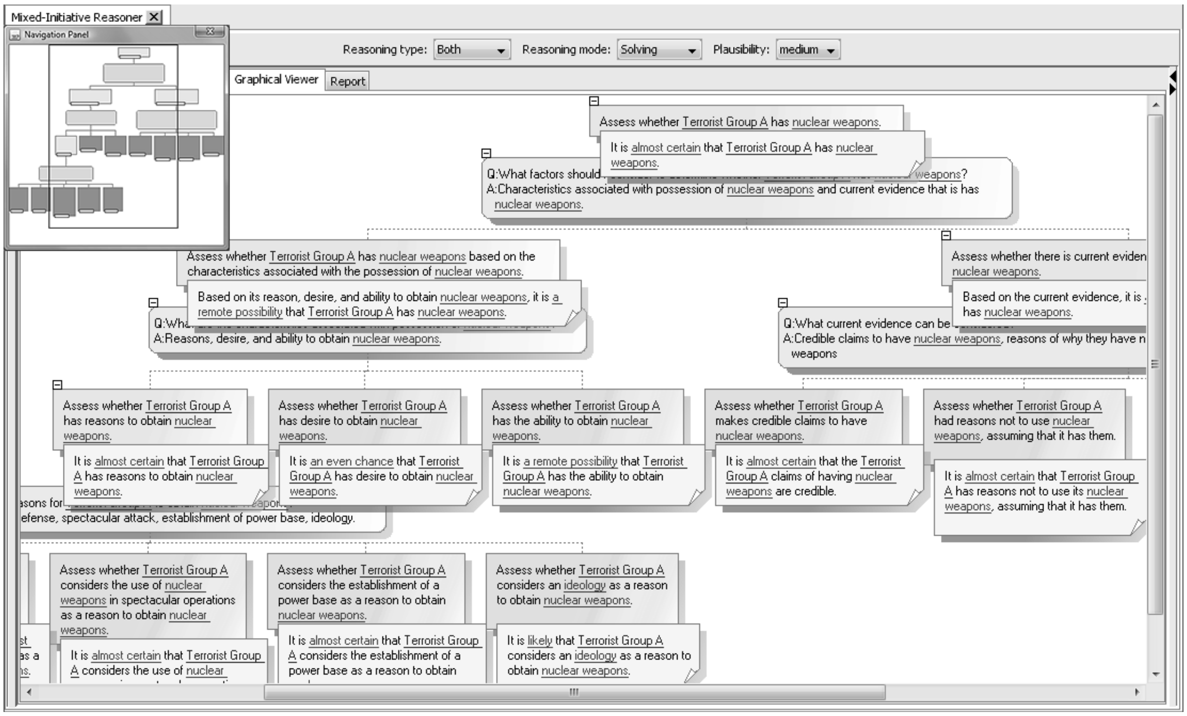

Figure 3: Hypothesis reduction and solution synthesis.

The reasoning trees generated by Disciple-LTA make very clear the analysis logic, what evidence was used and how, what assumptions have been made (see next section), and what is not known. They illustrate an ability of a decision-support system to generate a solution to a complex problem in a very transparent way, which is very natural for a human, but it is more precise and more detailed, and it is generated much faster. This allows the human to critically evaluate the reasoning process, accept parts of it, modify other parts, and produce a solution to the problem which s/he would consider her/his own.

In Intelligence Analysis (as well as in many other decision-making processes), the user has to consider the plausibility of the available information and has to deal with missing information. The analyst is therefore forced to assume different possibilities for the missing or uncertain information and evaluate the different outcomes in order to make a good decision about the situation. The next section illustrates some of the capabilities of Disciple-LTA that allows the user to define and analyze different what-if scenarios.

3. Assumption-Based What-If Scenarios

One approach to cope with incomplete and inconsistent information is to use multiple worlds reasoning [24], where multiple worlds (represented as knowledge bases) are generated from an initial one by adding assumptions, such that each world contains only non-contradictory assumptions. Each such world corresponds to a single what-if scenario. This approach may lead to a combinatorial explosion in the number of alternatives to be generated and analyzed. An alternative approach is to assign different probabilities or certainty factors to each uncertain fact and compute an overall probability/certainty factor for the suggested decisions [2]. However, this approach has the difficulty of eliciting all the required probabilities, and to express the influence of the missing knowledge on the overall result.

We have developed a different, more scalable approach, to cope with incomplete and inconsistent information. It is based on defining assumptions as solutions to sub-problems and not as facts in the ontology. The user of Disciple-LTA can select any problem from the generated reasoning tree and provide a solution to that problem, as an assumption. This approach is illustrated in Figure 4. The user invoked the Assumptions Assistant (the interface of which is shown in the right-hand side of Figure 4), then selected a problem from the reasoning tree generated by Disciple-LTA ("Assess whether Terrorist Group A has reasons not to use nuclear weapons, assuming that is has them."), and provided a solution for that problem ("It is unlikely that Terrorist Group A has reasons not to use nuclear weapons, assuming that is has them."). As a result, Disciple-LTA has immediately updated the reasoning three (which has over 1700 nodes) and synthesized a different solution to the initial hypothesis analysis problem ("It is likely that Terrorist Group A has nuclear weapons."). Together with the assumption, the user may also provide a justification which details the reasons for making that assumption.

To distinguish the solutions generated by the system from the assumptions made by the user, the assumptions are displayed with a yellow background. In addition, the assumption shown in Figure 4 also has a red border. This is because that particular assumption is challenged by the system which has obtained a different solution (see the right upper part of Figure 4).

The Assumptions Assistant allows the user to easily define, modify, delete, enable, disable, browse, view and search the assumptions used in a reasoning tree. More than one assumption may be associated with a given problem, but at most one of them can be enabled. This allows the user to easily experiment with different what-if scenarios by simply enabling a different set of assumptions which will be immediately used to update the current reasoning tree.

The user can use the Assumptions Assistant to hypothesize a solution for a problem that cannot be solved by the system, to change a solution generated by the system, or to experiment with different what-if scenarios.

As indicated in Section 1, in addition to solving problems in a natural way for a human, and to providing capabilities for dealing with incomplete and uncertain information (e.g. through assumptions), a good decision-support system should be able to collaborate with its user in a natural way, similar to how humans collaborate. The next section will discuss the natural mixed-initiative reasoning made possible by Disciple-LTA.

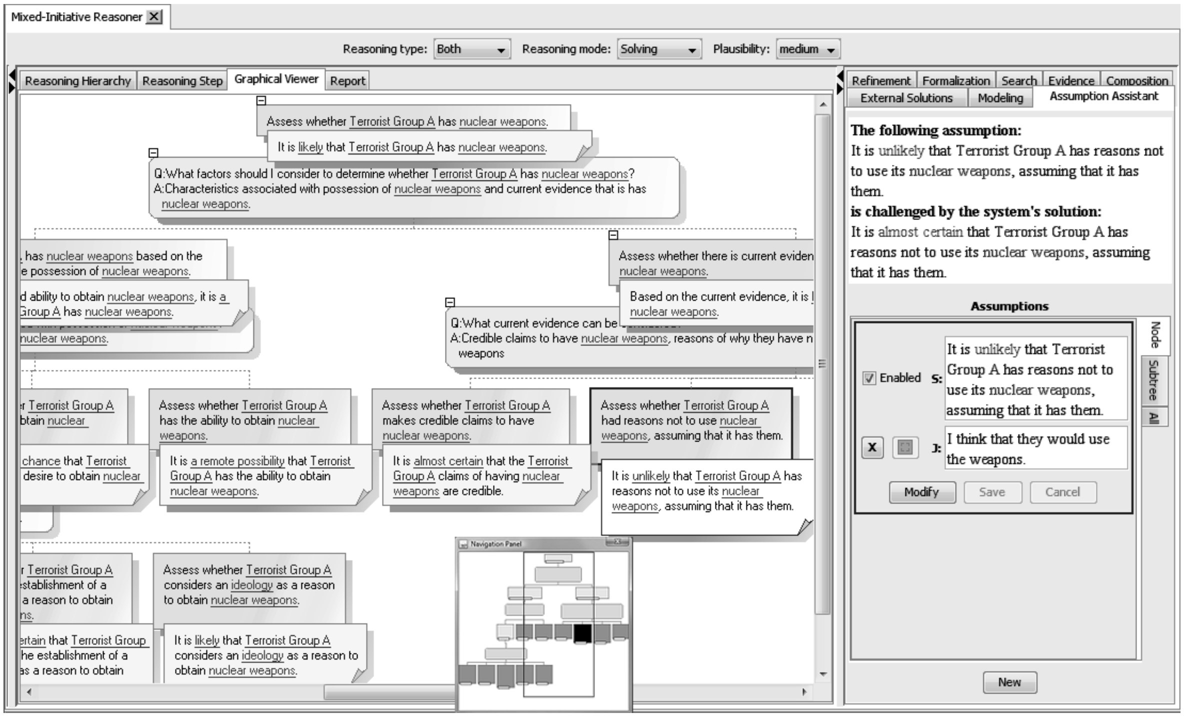

Figure 4: Assumptions-based analysis.

4. Mixed-Initiative Reasoning

Mixed-initiative reasoning is a type of collaboration between humans and automated agents that mirror the flexible collaboration between people. It is based on an efficient, natural interleaving of contributions by people and agents that is determined by their relative knowledge and skills and the problem-solving context, rather than by fixed roles, enabling each participant to contribute what it does best, at the appropriate moment [25, 26].

We will present the mixed-initiative reasoning enabled by Disciple-LTA, illustrating it with the collaboration between the Assumptions Assistant and the human analyst. This interaction is managed by an executable interaction model generated by Disciple from its interaction knowledge base. The interaction model indicates the next actions that the Assumption Assistant or the user can perform in a given state of the assumption definition process. Figure 5 shows a fragment of the interaction model where Disciple-LTA has to determine the next action for defining an assumption. It first checks the state of the interface to determine whether the Assumption Editor is displayed. If it is not displayed, then Disciple checks whether the user has clicked on the "New" button. If "No", it suggests the user to click on it. Otherwise it directs the Assumptions Assistant to display the assumption editor interface.

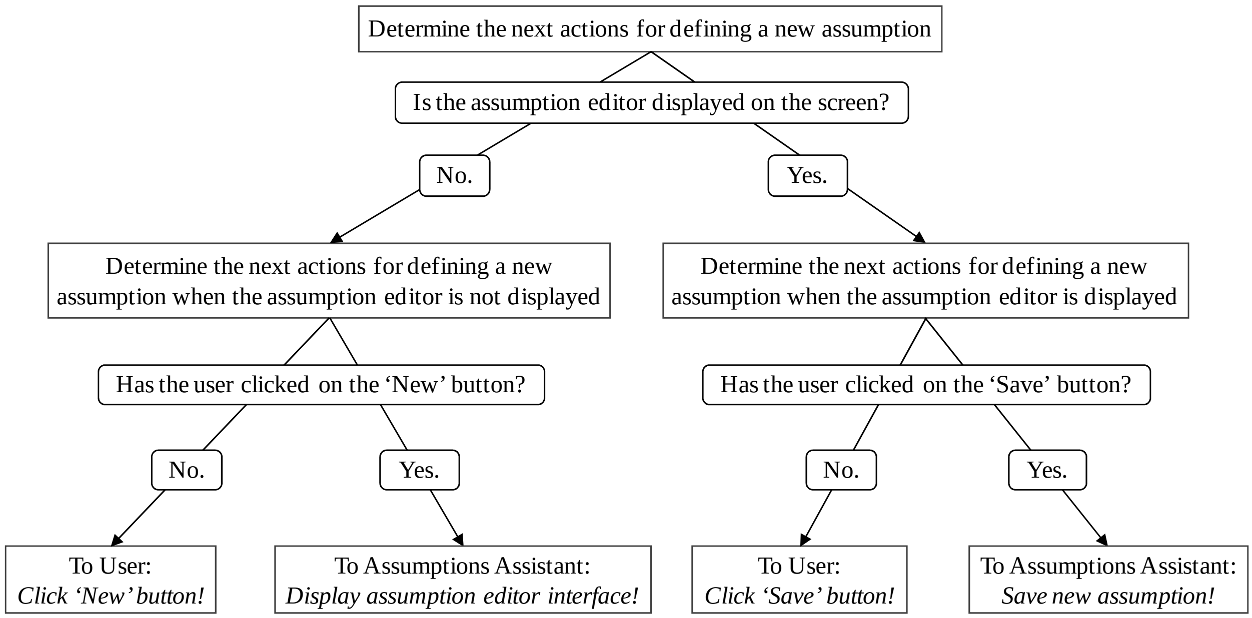

Figure 5. A fragment of the interaction model for defining a new assumption.

Disciple’s mixed-initiative interaction framework, which makes this type of interaction possible, is represented in Figure 6. The main component of the framework is the Mixed-Initiative Interaction Manager which handles the interactions between the various Disciple assistants and between each assistant and the user. The manager includes an interaction knowledge base, an interaction engine, a task agenda and a learning engine.

The interaction knowledge base contains the interaction model, represented as tasks and reduction rules, and the problem solving state, represented as instances and facts. The interaction engine uses the interaction model generated for the current state to obtain the set of actions that can be performed at that time by the assistants and by the user. The generated actions are posted on the task agenda and then forwarded to the appropriate Disciple assistants (such as the Assumptions Assistant). The actions that can be executed by the user are displayed by each assistant interface in one or several graphical controls. For example, an action may correspond to the ‘New’ button for creating a new assumption. Each performed action (by the user or by an assistant) leads to an updated state and updated actions on the task agenda. The responsibility for updating the state knowledge base belongs to the Disciple assistant that executed an action.

Using a knowledge base to represent the interaction knowledge and generate interaction models allows the development and maintenance of the interaction flaw by a knowledge engineer rather than a system developer. Moreover, it allows the learning of interaction rules during the actual use of Disciple to better fit the preferences and the needs of the user.

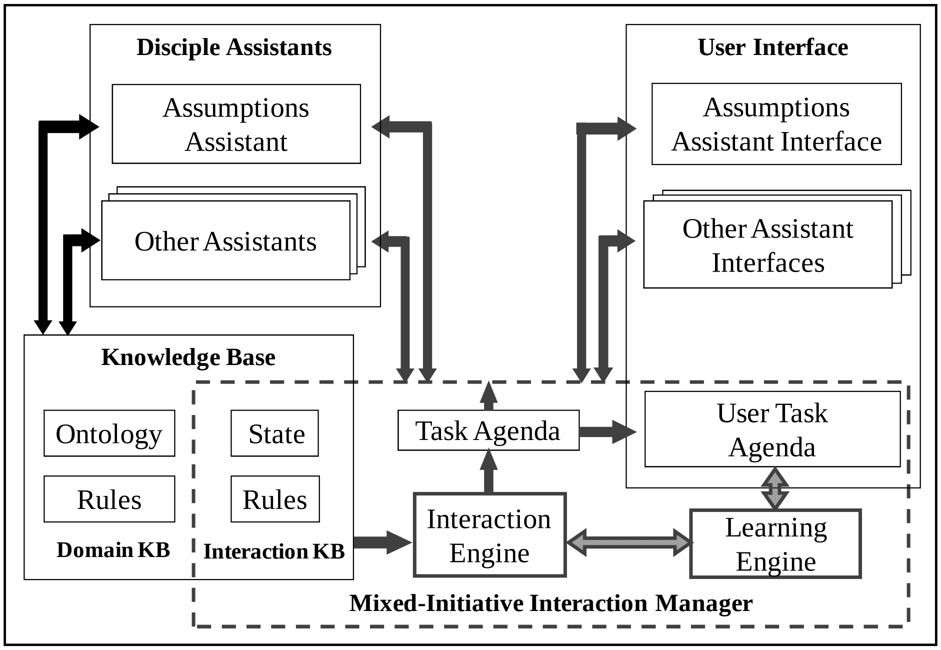

Figure 6. Disciple's mixed-initiative framework.

5. Final Remarks

Disciple-LTA has been evaluated in Spring 2007 as part of the US Army War College course entitled "Military Applications of Artificial Intelligence: Intelligence Analysis." During this course, 7 high-ranking military officers experimented with Disciple-LTA as a learning system, as a tutoring system, and as a decision making assistant, and then assessed its various capabilities. Figure 7 shows some of the assessments related to the decision-making capabilities discussed in this paper. In particular they show that all the experts agreed or strongly agreed that Disciple-LTA is easy to use, that its reasoning logic is easy to understand, and that the use of the Assumptions Assistant is a good approach to hypothesis analysis with incomplete information and to the investigation of the what-if scenarios.

The experimental results show also that all the discussed capabilities could be further improved. One direction of improvement is the development of a more abstract, simplified representation of the reasoning process and the associated reasoning trees, highlighting the critical steps, but also allowing the user to drill down for details, if needed. Another direction of improvement relates to the management of assumptions, such as the ability to define groups of assumptions that could be collectively enabled or disabled, to suggest new assumptions by analogy, or to compare competing what-if analyses. An improved ability to learn mixed-initiative patterns from its user would allow Disciple-LTA to behave as a more natural extension of the user's reasoning capabilities.

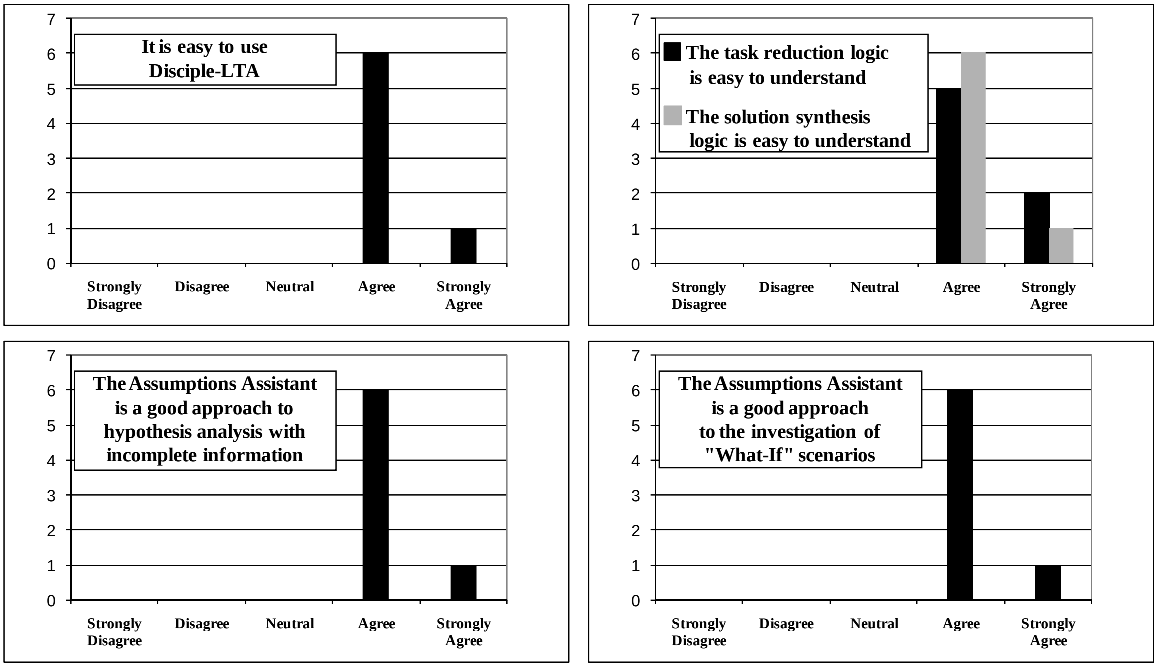

Figure 7. Sample evaluation results.

Acknowledgements

This paper is dedicated to Florin G. Filip, in recognition of his contributions to decision-support systems, on the occasion of his 60th birthday.

This material is based on research performed in the Learning Agents Center and sponsored by several organizations, including the Air Force Research Laboratory under agreement number FA8750-04-1-0257, the Air Force Office of Scientific Research under grant number F9550-07-1-0268, the National Science Foundation under grant number 0610743 and the US Army War College. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained therein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory, the Air Force Office of Scientific Research, the National Science Foundation, the US Army War College, or the U.S. Government.

REFERENCES

1. SIMON, H.A., **The Sciences of the Artificial (3rd ed.)**, The MIT Press, 1996.

2. TURBAN, E., **Decision Support and Expert Systems: Management Support Systems** (2nd ed.), Macmillan Publishing Company, New York, 1990.

3. FILIP, F.G., **Systems Analysis and Expert Systems Techniques for Operative Decision Making**, in A. Sydow (ed.), Computational Systems Analysis: Topics and Trends, Elsevier Sci. Publishers, Amsterdam, 1992, pp. 285-305.

4. FILIP, F.G., BARBAT, B., **Industrial Informatics: New Paradigms and Applications**, Ed. Tehnica, Bucharest, 1999.

5. GUPTA, J.N.D., FORGIONNE, G.A., MORA, M., (eds.), **Intelligent Decision-making Support Systems: Foundations, Applications and Challenges**, Springer-Verlag, London, 2006.

6. PHILLIPS-WREN, G.A., JAIN, L.C., (eds.), **Intelligent Decision Support Systems in Agent-Mediated Environments**, IOS Press, Amsterdam, 2005.

7. TUROFF, M., **Design of Interactive Systems,** in Emergency Management Information Systems Tutorial, The Hawaii International Conference on System Sciences, HICSS-40, Hawaii, 3 January 2007.

8. POMEROL, J.C., ADAM, F., **On the Legacy of Herbert Simon and his Contribution to Decision-making Support Systems and Artificial Intelligence**, in Jatinder N.D. Gupta, Giusseppi A. Forgionne and Manuel Mora T. (eds.), Intelligent Decision-making Support Systems: Foundations, Applications and Challenges, Springer-Verlag, London, 2006, pp. 25-43.

9. FILIP, F.G. (coordinator), **Informational Society - Knowledge Society**, Ed. Expert, Bucuresti, 2001.

10. ROONEY, D., HEARN, G., NINAN, A., **Handbook on the Knowledge Economy**, Edward Elgar Publishing, Inc., 2005.

11. TECUCI, G., BOICU, M., MARCU, D., BOICU, C., BARBULESCU, M., AYERS, C., CAMMONS, D., **Cognitive Assistants for Analysts**, in John Auger, William Wimbish (eds.), Proteus Futures Digest: A Compilation of Selected Works Derived from the 2006 Proteus Workshop, Joint publication of the National Intelligence University, Office of the Director of National Intelligence, and US Army War College Center for Strategic Leadership, 2007, pp. 303-329.

12. FILIP, F.G., **Creativity and Decision Support System**, Studies and Researches in Computer and Informatics, 1(1), 1989, pp. 41-49.

13. TECUCI, G., **An Environment for Developing Instructible Expert Systems for Computer Aided Design**, PhD Thesis, Polytechnic Institute of Bucharest, December 1988.

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

15. BOICU, M., **Modeling and Learning with Incomplete Knowledge**, PhD Thesis in Information Technology, Learning Agents Laboratory, The Volgenau School of Information Technology and Engineering, George Mason University, Fairfax Virginia, 2002.

16. TECUCI, G., BOICU, M., BOWMAN, M., MARCU, D., with a commentary by BURKE, M., **An Innovative Application from the DARPA Knowledge Bases Programs: Rapid Development of a High Performance Knowledge Base for Course of Action Critiquing**, in AI Magazine, 22 (2), AAAI Press, Menlo Park, California, 2001, pp. 43-61.

17. TECUCI, G., BOICU, M., BOICU, C., MARCU, D., STANESCU, B., BARBULESCU, M., **The Disciple-RKF Learning and Reasoning Agent**, in Computational Intelligence, *21 (4)*, 2005, pp. 462-479.

18. TECUCI, G., BOICU, M., HAJDUK, T., MARCU, D., BARBULESCU, M., BOICU, C., LE, V., **A Tool for Training and Assistance in Emergency Response Planning**, in Proceedings of the Hawaii International Conference on System Sciences, HICSS-40, Hawaii, January 3-6, 2007.

19. NILSSON, N.J., **Problem Solving Methods in Artificial Intelligence**, NY: McGraw-Hill, 1971.

20. POWELL, G.M., SCHMIDT, C.F., **A First-order Computational Model of Human Operational Planning**, in CECOM-TR-01-8, US Army CECOM, Fort Monmouth, NJ, 1988.

21. BOICU M., TECUCI G., MARCU D., **Mixed-Initiative Assistant for Modeling Expert's Reasoning,** in D. Aha, G. Tecuci (eds.), Proceedings of the AAAI-05 Fall Symposium on Mixed-Initiative Problem-Solving Assistants, Arlington, Virginia, November 4-6, 2005, pp.7-13.

22. BOICU, C., **An Integrated Approach to Rule Refinement for Instructible Knowledge-Based Agents,** PhD Thesis in Computer Science, Learning Agents Laboratory, The Volgenau School of Information Technology and Engineering, George Mason University, Fairfax Virginia, 2006.

23. SCHUM, D.A., **The Evidential Foundations of Probabilistic Reasoning,** Northwestern University Press, 2001.

24. MENZIES, T., **Verification and Validation and Artificial Intelligence**, in Foundations '02: A Verification and Validation Workshop, Kossiakoff Conference and Education Center, Johns Hopkins U. Applied Physics Lab Laurel, Maryland USA, October 22-23, 2002.

25. TECUCI, G., BOICU, M., COX, M.T., (guest editors), **AI Magazine, Special Issue on Mixed-Initiative Assistants**, 28 (2), 2007.

26. TECUCI, G., BOICU, M., COX, M.T., **Seven Aspects of Mixed-Initiative Reasoning: An Introduction to the Special Issue on Mixed-Initiative Assistants**, AI Magazine, 28 (2), 2007, pp. 11-18.