in Proceedings of the 12th International Conference on Machine Learning and Applications (ICMLA'13), Miami, Florida, December 4-7, 2013.

How Learning Enables Intelligence Analysts to Rapidly Develop Practical Cognitive Assistants

Gheorghe Tecuci, Mihai Boicu, Dorin Marcu, David Schum
Learning Agents Center, Volgenau School of Engineering, George Mason University, Fairfax, VA 22030, USA

***Abstract*—This paper overviews an end-to-end learning-based approach to the rapid development of practical cognitive assistants for intelligence analysis. A learning agent shell has been trained by a knowledge engineer with general evidence-based reasoning knowledge for intelligence analysis. This agent is further trained by an expert analyst how to analyze complex hypotheses from a given intelligence analysis domain. The resulting cognitive assistant is used by a typical analyst to rapidly analyze hypotheses from agent's area of expertise. During its use, the agent continues to learn reasoning patterns from its user. This approach has been implemented and practical agents have been developed and used. This is a significant application of machine learning to agents development in intelligence analysis that can be generalized to many other domains involving evidence-based reasoning, including medicine, law, and science.**

***Keywords*—learning agent shell; intelligence analysis; rule learning; ontology; evidence-based reasoning; cognitive assistant**

I. INTRODUCTION

For many years we have research a theory, methodology, and tools for the development of knowledge-based cognitive assistants that: (1) learn complex problem solving expertise directly from subject matter experts; (2) support experts and non-experts in problem solving; and (3) teach their problem solving expertise to students.

The investigated approach relies on developing a powerful learning agent shell that can be taught by a subject matter expert (who does not have computer science or knowledge engineering experience) in ways that are similar to how the expert would teach a student or an apprentice, by explaining problem solving examples to it, and by supervising and correcting its problem solving behavior. Because the resulting agent learns to replicate the problem-solving behavior of its human expert, we have called it a Disciple agent ([1], [2], [3], [4], [5], [6]).

The long term goal of the Disciple approach is to contribute to a new revolution in the use of computers by enabling typical computer users to develop their own cognitive assistants. Thus, non-computer scientists will no longer be only users of generic programs developed by others (such as word processors or Internet browsers), as they are today, but also agent developers themselves. They will be able to train their personal Disciple assistants to help them with their increasingly complex tasks in the knowledge society, which should have a significant beneficial impact on their work and life. This goal is consistent with the Semantic Web vision of enabling typical users to author web content that can be understood by automated agents [8]. Very recently, Bill Gates has also stressed the great potential and importance of the software assistants [7].

In this paper we overview a significant advancement of the Disciple approach to the development of cognitive assistants. The basic idea is to customize the general Disciple learning agent shell to a specific application domain, such as Intelligence Analysis. This enables a *knowledge engineer* to incorporate general problem-independent knowledge for hypotheses analysis into the Disciple shell. Thus, when such an agent is further trained by an *expert analyst*, it only needs to learn hypotheses-specific knowledge, which significantly simplifies the agent development task. After that the agent can be used by a *typical analyst* to analyze specific hypotheses based on evidence. At this point, however, it can use even simpler teaching and learning methods, which are enough because the agent has acquired much of the necessary knowledge in the previous stages.

In the next section we introduce the highly complex Intelligence Analysis (IA) domain. After that we present the general knowledge that we have taught the Disciple learning agent shell, evolving it into a learning agent shell for intelligence analysis. Then we overview the methodology for building a Disciple cognitive assistant for a specific IA application, emphasizing the role of learning in the different stages of this methodology. We continue with an introduction of the developed cognitive assistants for IA. Then we discuss the applicability of this approach to other domains requiring evidence-based reasoning, such as medicine, law, and science. Because the covered topics are many and complex, we will only present the general ideas and approaches, referring to other papers for details.

II. INTELLIGENCE ANALYSIS

Within the framework of the scientific reasoning, we have developed a *computational theory of intelligence analysis* by viewing it as ceaseless discovery of *evidence*, *hypotheses*, and *arguments* in a non-stationary world, involving collaborative processes of *evidence in search of hypotheses*, *hypotheses in search of evidence*, and *evidentiary testing of hypotheses* ([9], [10]). Fig. 1 is an abstract illustration of this astonishingly complex process, in the context of predictive analysis, where the likeliness of future events is assessed. Suppose we obtain evidence E* (e.g., a prominent U.S. political leader gave a talk supporting the development of wind power). This suggests, through a chain of abductive reasoning, that the hypothesis H (the United States will be a world leader in wind power within the next decade) might be true: Because of E* it is possible that

F might be true. Therefore G might be true. Therefore H might be true. The problem with drawing this conclusion, however, is that there are other hypotheses that also explain E* (e.g., F', G', H'). To conclude H, we would need to assess each competing explanatory hypothesis, based on additional evidence, showing that F, G, and H are more likely than their competing hypotheses. Suppose that we have shown that F and G are more likely than their competing hypotheses. Now we have to assess H, … , H'. To assess H, we need additional evidence which is obtained by successively decomposing H into simpler and simpler hypotheses, as shown in the middle of Fig. 1. H would be true if G and M would be true. Then M would be true if N, P, and Q would be true. But if N would be true, then we would need to observe evidence E_n*. So we look for E_n* and we may or may not find it. This is the process of hypotheses in search of evidence that guides the evidence collection task. Then we use the identified evidence to assess the likeliness of H. In the developed computational theory of intelligence analysis [10], hypotheses assessment is based on a combination of ideas from the Baconian and Fuzzy systems of probability ([11], [12]), and the likeliness of a hypothesis may have one of the following ordered values:

no support < likely < very likely < almost certain < certain

The likeliness of an upper-level hypothesis (e.g., H) is obtained from the likeliness of its sub-hypotheses (i.e., G and M) by using min or max Baconian and Fuzzy combination functions, depending on whether the sub-hypotheses G and M represent necessary and sufficient conditions for the hypothesis H, sufficient conditions, or just indicators.

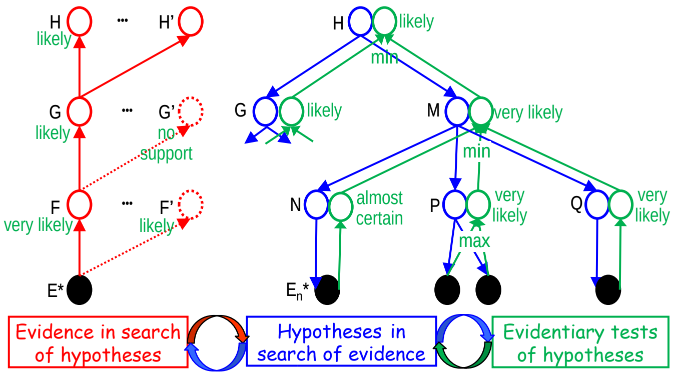


Fig. 1. Scientific reasoning framework of intelligence analysis.

III. Learning Agent Shell for Intelligence Analysis

A learning assistant for intelligence analysis would support an analyst in performing the reasoning from Fig. 1, while continuing to learn from their joint analysis process. Such a learning assistant is developed by training the Disciple learning agent shell for intelligence analysis. The overall architecture of this shell is shown in Fig. 2. Notice the hierarchical organization of its knowledge bases (KB). At the top level is the general knowledge base for intelligence analysis (IA KB), containing knowledge applicable to the evidence-based analysis of any type of intelligence hypothesis, from any domain. Under it, and inheriting from it, are domain-specific knowledge bases. Each such Domain KB contains knowledge specific to a particular type of IA problems, such as predictive analysis related to energy sources, or assessments related to the current production of weapons of mass destruction by various actors. Under each Domain KB there are several Scenario KBs, each corresponding to an instance of a problem pattern from that domain such as: "Assess whether the United States will be a world leader in wind power within the next decade." This Scenario KB will contain specific knowledge about the United States, as well as items of evidence to make the corresponding analysis. The actual analysis will be done by using this knowledge as well as more general knowledge inherited from the corresponding Domain KB and from the IA KB.

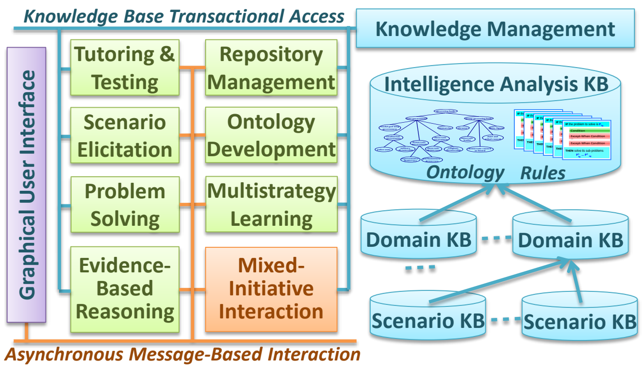
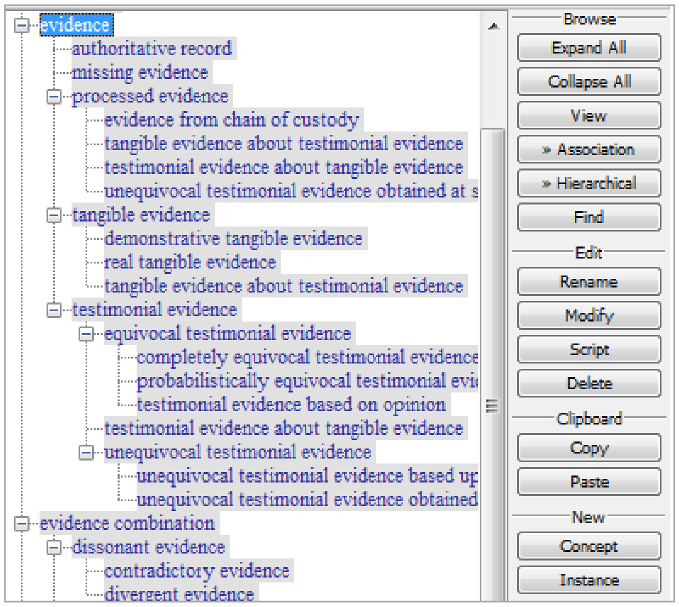


Fig. 2. Learning agent shell for intelligence anlaysis.

Each of these knowledge bases is structured into an ontology of concepts and a set of general problem solving rules expressed with these concepts. The rules are learned from specific examples of reasoning steps, by using the ontology as a generalization hierarchy [5]. The learning agent shell for IA is obtained by training the Disciple learning agent shell with general intelligence analysis knowledge from the computational theory, resulting in the development of the IA KB. The IA KB contains both a general ontology and a set of general reasoning rules which are necessary for any Disciple agent for intelligence analysis, as we will briefly present in this section. For example, Fig. 3 shows a general ontology of evidence which, among others, distinguishes between different types of tangible and testimonial evidence [13].

Learned general rules include those for directly assessing a hypothesis based on evidence. As shown in Fig. 4, these rules automatically reduce the assessment of any hypothesis P to assessments based on favoring and disfavoring evidence and, further down, to the assessment of the *relevance* and the *believability* of each item of evidence with respect to P. Once these assessments are made, they are combined, from bottom-up, to obtain the *inferential force* of all the items of evidence on P, which results in the likeliness of P. These assessments are probabilistic because the evidence is always *incomplete*, usually *inconclusive*, frequently *ambiguous*, commonly *dissonant*, and has various degrees of *believability* [9].

Learned general rules also include those for assessing the believability of different types of evidence. For example, to assess the *believability* of an item of testimonial evidence (i.e. a statement provided by a person), one needs to assess the *competence* and *credibility* of that person. Person's competence is assessed by assessing his/her *access* and *understandability*, while his/her credibility is assessed by assessing his/her *veracity*, *objectivity*, and *observational sensitivity*. Other learned believability rules correspond to mixed evidence, such as tangible evidence about testimonial evidence, or evidence obtained through a chain of custody such as when a person describes the observation performed by another person [14].



- evidence
  - authoritative record
  - missing evidence
  - processed evidence
    - evidence from chain of custody
    - tangible evidence about testimonial evidence
    - testimonial evidence about tangible evidence
    - unequivocal testimonial evidence obtained at s
  - tangible evidence
    - demonstrative tangible evidence
    - real tangible evidence
    - tangible evidence about testimonial evidence
  - testimonial evidence
    - equivocal testimonial evidence
      - completely equivocal testimonial evidence
      - probabilistically equivocal testimonial evi
      - testimonial evidence based on opinion
    - testimonial evidence about tangible evidence
    - unequivocal testimonial evidence
      - unequivocal testimonial evidence based up
      - unequivocal testimonial evidence obtained
- evidence combination
  - dissonant evidence
    - contradictory evidence
    - divergent evidence

Fig. 3. Ontology fragment showing various types of evidence.

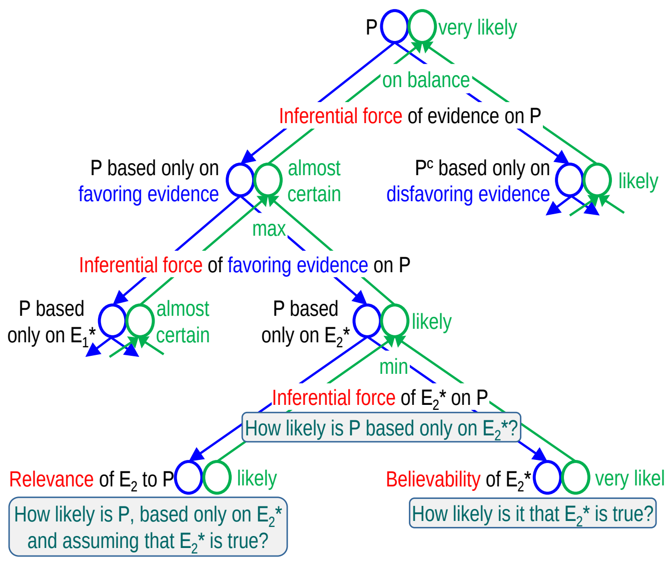

P
very likely
on balance
Inferential force of evidence on P
P based only on favoring evidence
almost certain
P^c based only on disfavoring evidence
likely
max
Inferential force of favoring evidence on P
P based only on E$_1$*
almost certain
P based only on E$_2$*
likely
min
Inferential force of E$_2$* on P
How likely is P based only on E$_2$*?
Relevance of E$_2$ to P
likely
Believability of E$_2$*
very likely
How likely is P, based only on E$_2$* and assuming that E$_2$* is true?
How likely is it that E$_2$* is true?

Fig. 4. The relevance, believability, and inferential force of evidence.

The important point here is that the Disciple learning agent shell has been taught a significant amount of general intelligence analysis knowledge, transforming it into a customized learning agent shell for intelligence analysis.

IV. Agent Development Methodology

Fig. 5 shows the main stages of the learning-based methodology of evolving the learning agent shell for intelligence analysis into a cognitive assistant for a particular IA domain, such as predictive analysis on energy sources.

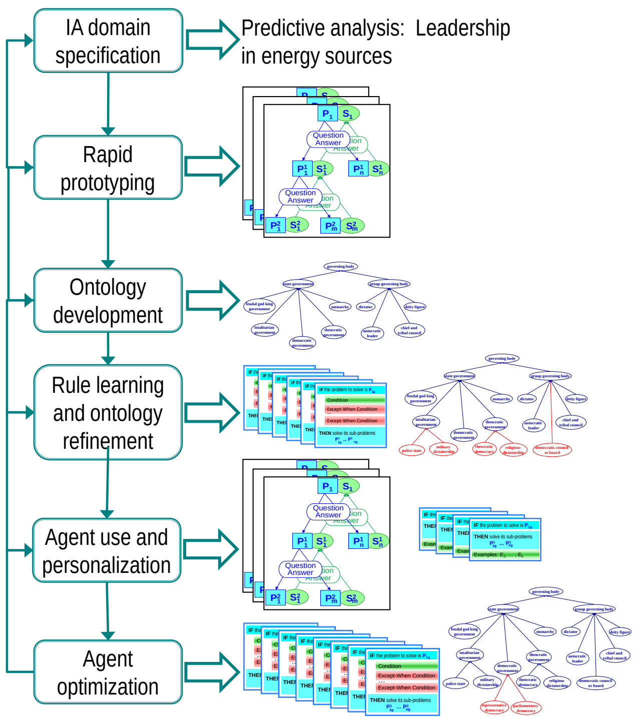
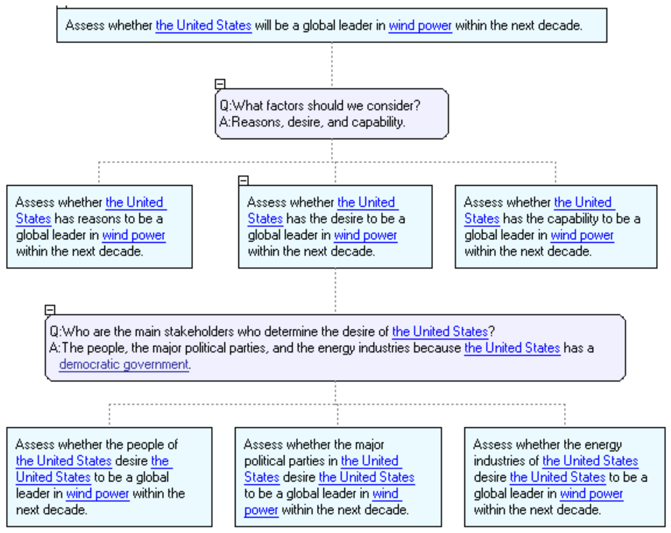

IA domain specification
Predictive analysis: Leadership in energy sources
Rapid prototyping
Ontology development
Rule learning and ontology refinement
Agent use and personalization
Agent optimization

Fig. 5. Main stages of the agent development methodology.

The first stage is the specification of the IA domain during which a knowledge engineer and a subject matter expert define the types of hypotheses to be analyzed with the agent. For example, the domain might be predictive analysis of what actors will be the world leaders in different types of energy sources, within a certain period of time. Another domain might be assessing whether a certain actor is currently pursuing a certain type of weapon of mass destructions.

The second stage is rapid prototyping, where the knowledge engineer is supporting the subject matter expert to develop argumentation structures for specific but representative hypotheses. A general example of an argumentation structure is the one from the right side of Fig.1. A specific example is shown in Fig.6. In such an argumentation structure, a complex hypothesis is assessed by: (1) Successively reducing its assessment, from top-down, to the assessment of simpler and simpler hypotheses; (2) Assessing the simplest hypotheses based on evidence, as was illustrated in Fig. 4; and (3) Successively combining, from bottom-up, the assessments of the simpler hypotheses, until the assessment of the top-level hypothesis is obtained.

Notice in Fig. 6 that the reduction of each hypothesis analysis problem is guided by an introspective question/answer pair. Consider the problem "Assess whether the United States has the desire to be a global leader in wind power within the next decade." The expert asks herself "Who are the main stakeholders who determine the desire of the United States?" The answer "The people, the major political parties, and the energy industries because the United States has a democratic government" guides the expert to reduce the problem to three simpler problems, as shown at the bottom part of Fig. 6. Notice that the question/answer pair also includes the explanation of the reduction ("because the United States has a democratic government") which will greatly facilitate the learning of a general reduction rule from this step alone, as will be discussed later in this section.



Assess whether the United States will be a global leader in wind power within the next decade.

Q:What factors should we consider?
A:Reasons, desire, and capability.

Assess whether the United States has reasons to be a global leader in wind power within the next decade.

Assess whether the United States has the desire to be a global leader in wind power within the next decade.

Assess whether the United States has the capability to be a global leader in wind power within the next decade.

Q:Who are the main stakeholders who determine the desire of the United States?
A:The people, the major political parties, and the energy industries because the United States has a democratic government.

Assess whether the people of the United States desire the United States to be a global leader in wind power within the next decade.

Assess whether the major political parties in the United States desire the United States to be a global leader in wind power within the next decade.

Assess whether the energy industries of the United States desire the United States to be a global leader in wind power within the next decade.

Fig. 6. Inquiry-driven hypothesis analysis.

The next stage is that of ontology development. The guiding question is: What are the domain concepts, relationships and instances that would enable the agent to automatically generate the reasoning trees developed during rapid prototyping?

From each reasoning step the knowledge engineer identifies the ontology elements mentioned in it. For example, the reasoning step discussed above suggests that the Domain KB should include the objects and the relationships shown in Fig. 7. Such semantic network fragments represent a specification of the needed ontology. In particular, this fragment suggests the need for a hierarchy of government types (democratic, totalitarian, etc.), a hierarchy of power types (wind, solar, etc.), and the feature has as government (with state as domain and government as range).

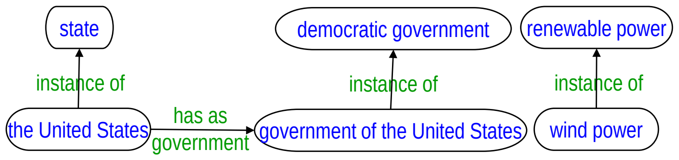
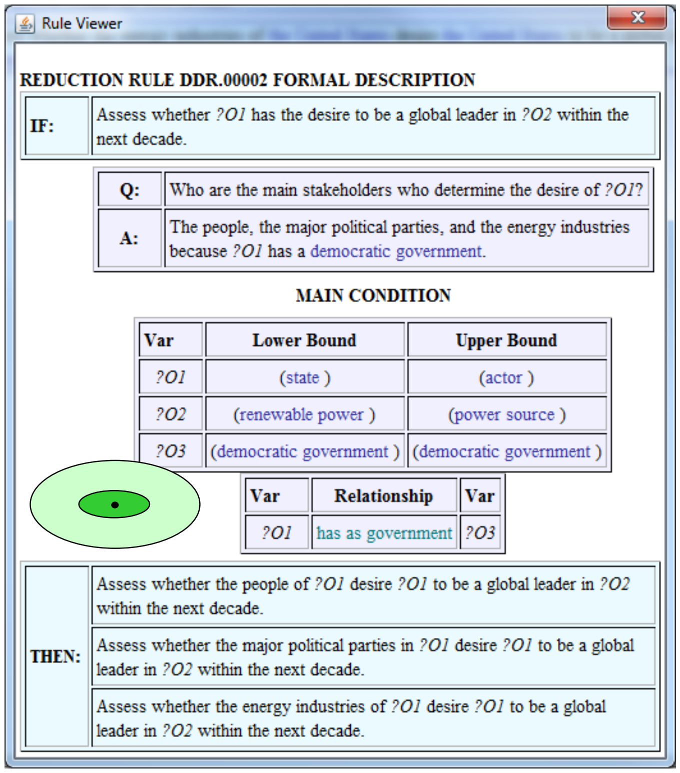


Fig. 7. Network fragments as ontology specification and explanation.

Based on such specifications, and using the ontology development tools of the Disciple shell, the knowledge engineer develops an ontology that is as complete as possible by importing concepts and relationships from previously developed ontologies, including those on the semantic web [8].

The next stage in agent development is that of rule learning and ontology refinement. From each problem reduction step of a reasoning tree developed during rapid prototyping the agent will learn a general problem reduction rule, by using the ontology as a generalization hierarchy [5], [6]. To illustrate, the rule learned from the bottom reduction in Fig. 6 (discussed above), is shown in Fig. 8. This is an IF-THEN reduction rule with a plausible version space applicability condition (the MAIN CONDITION). The rule pattern is obtained by generalizing each instance and constant in the reduction step to a variable (e.g. the United States is generalized to ?O1). The lower bound of the MAIN CONDITION is obtained through a minimal generalization of the semantic network fragment from Fig. 7 (which represents the formal explanation of why the reduction step is correct). The upper bound is obtained through a maximal generalization. During rule refinement (discussed below) the two bounds will converge toward one another and toward the exact applicability condition which will assure that the rule will only generate correct reductions.



REDUCTION RULE DDR.00002 FORMAL DESCRIPTION

IF:	Assess whether ?O1 has the desire to be a global leader in ?O2 within the next decade.

Q:	Who are the main stakeholders who determine the desire of ?O1?
A:	The people, the major political parties, and the energy industries because ?O1 has a democratic government.

MAIN CONDITION

Var	Lower Bound	Upper Bound
?O1	(state)	(actor)
?O2	(renewable power)	(power source)
?O3	(democratic government)	(democratic government)

Var	Relationship	Var
?O1	has as government	?O3

THEN:	Assess whether the people of ?O1 desire ?O1 to be a global leader in ?O2 within the next decade.
	Assess whether the major political parties in ?O1 desire ?O1 to be a global leader in ?O2 within the next decade.
	Assess whether the energy industries of ?O1 desire ?O1 to be a global leader in ?O2 within the next decade.

Fig. 8. Rule learned from a reduction example.

What the rule in Fig. 8 says is the following:

IF the problem to be solved is "Assess whether ?O1 has the desire to be a global leader in ?O2 within the next decade."

and either the lower or the upper bound condition is satisfied (i.e., ?O1 is a state that has as government ?O3 which is a democratic government, and ?O2 is renewable power, or, …)

THEN solve the three sub-problems: "Assess whether the people of ?O1 desire ?O1 to be a global leader in ?O2 within the next decade", "Assess whether the major political parties in ?O1 …", … .

During this phase the expert analyst teaches the agent to solve other similar problems. She instantiates a learned problem pattern, such as "Assess whether China will be the world leader in solar power within the next decade", and the agent automatically generates the reasoning tree, by applying the learned rules. The analyst then critiques agent's reasoning, guiding it in refining the rules [5]. Incorrect reductions and their explanations lead to the specialization of the rule, either by specializing the upper bound of the MAIN CONDITION, or by learning the plausible version space for a new EXCEPT-WHEN CONDITION (which should not be satisfied for the rule to be applicable), or by generalizing the lower bound of an existing EXCEPT-WHEN CONDITION, or by adding a negative exception when none of the above operations is possible. Correct reductions lead to the generalization of the rule, either by generalizing the lower bound of the MAIN CONDITION, or by specializing the upper bound of one or several EXCEPT-WHEN CONDITIONs, or by adding a positive exception when none of the above operations is possible.

Notice in the representation of a learned rule the same idea as in the semantic web: We have two representations of the rule's applicability condition, a natural language expression for human use (represented by the question/answer pair), and a formal representation for automatic reasoning by the agent (represented by the MAIN CONDITION).

Now the assistant is ready to be used by typical analysts, as part of the next phase: Agent use and personalization. Typically, the analyst will specify the hypothesis to analyze (e.g., Assess whether United Kingdom will be a world leader in wave power within the next decade) by simply instantiating a corresponding pattern. Then the agent will automatically generate a reduction tree like the one in Fig. 6, by automatically applying the learned reduction rules. This tree reduces the top level hypothesis to elementary hypotheses to be directly assessed based on evidence. The analyst will then have to search the Internet and other repositories for evidence, and attach each item of evidence to the hypothesis to which it is relevant. As a result, the agent will automatically reduce the elementary hypotheses as indicated in Fig. 4 (where P corresponds to an elementary hypothesis). Next the analyst has to assess the relevance and the believability of each item of evidence (see the bottom assessments in Fig. 4), and the agent automatically computes the inferential force of evidence, by applying the corresponding synthesis functions. This will result in the likeliness of the top-level hypothesis. The analyst may also direct the agent to perform a deeper believability assessment of specific items of evidence, when they are critical to the final result [13].

So when does the additional learning take place? It may be the case that the agent does not know how to reduce a specific hypothesis. Then the analyst needs to indicate its reduction, either to elementary hypotheses or to known hypotheses. As a result the agent will automatically learn reduction patterns which are just like the rule in Fig. 6, except that they do not have applicability conditions. Another important difference between a learned rule and a learned pattern concerns their use in problem solving. A learned rule is automatically applied, while a learned pattern is proposed to the user who may decide to select it and instantiate it appropriately. If this is done, the corresponding instances will be added to the pattern. Thus this type of learning is much simpler for the analyst who only needs to specify, in natural language, the reduction of hypotheses to simpler hypotheses.

Periodically the agent can undergo an optimization phase which is the last phase in Fig. 5. During this phase, a knowledge engineer and an expert analyst will review the patterns learned from the typical analyst, will learn corresponding rules from them, and will correspondingly refine the ontology. The current version of the Disciple shell reapplies its rule and ontology learning methods to do this. However, we plan to develop improved methods for the situation where a learned pattern has more than one set of instances, because each represents a different example of the rule to be learned.

V. THE TIACRITIS AGENTS

The learning agent shell for intelligence analysis described in this paper has been implemented and the described methodology has been employed to develop practical agents for intelligence analysis. These agents are known in the Intelligence Community as TIACRITIS agents [15], [16]. This is an acronym for Teaching Intelligence Analysts Critical Thinking Skills, and reflects the primary goal of these agents. Their use is supported by a textbook which contains numerous case studies of analysis, enabling a quite unique learning by doing approach [16]. The types of hypotheses that TIACRITIS agents have been taught to analyze include: A non-state actor has nuclear weapons; A state actor is pursuing a nuclear program for military purposes; A state actor is pursuing a nuclear program for economic purposes; There is an ambush threat at a certain location; A terrorist organization will set-off a dirty bomb in a certain location; A government will crack down on its opposition; The government of a country supports the insurgency in another country, A professor would be a good advisor for a student; A website is believable, etc.

The use of the TIACRITIS agents in military and intelligence organizations represent a significant validation of the described research.

VI. COGNITIVE ASSISTANTS FOR EVIDENCE-BASED REASONING

Intelligence analysis, introduced in Section II, is an example of evidence-based reasoning. Consider again the reasoning framework from Fig. 1. The same reasoning framework applies to other evidence-based reasoning domains, as illustrated in Fig. 9 and explained in the following.

In medicine, a doctor makes observations related to a patient's complaints and hypothesizes possible illnesses that would explain them. She then performs various medical tests that provide further evidence which is used to assess the likeliness of the considered illnesses.

In law, an attorney makes observations in a criminal case and seeks to generate hypotheses in the form of charges that seem possible in explaining these observations. Then, assuming that a charge is justified, attempts are made to deduce further evidence bearing on it. Finally, the obtained evidence is used to prove the charge.

In forensics, observations made at the site of an explosion in a power plant lead to the formulation of several possible causes. Analysis of each possible cause leads to the discovery of new evidence that eliminates or refines some of the causes, and may even suggest new ones. This cycle continues until enough evidence is found to determine the most likely cause.

Scientists from various domains, such as physics, chemistry, or biology, may recognize this framework as a formulation of a basic scientific method.

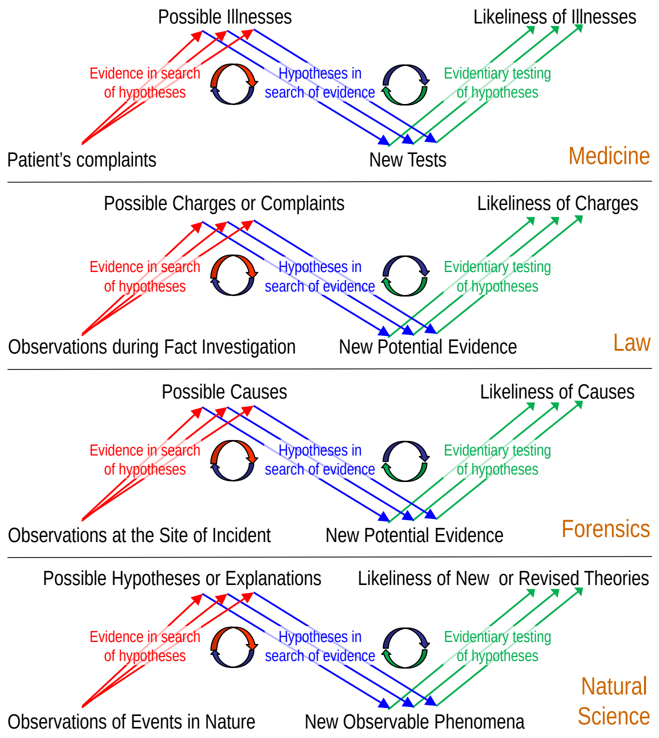

Possible Illnesses
Likeliness of Illnesses
Evidence in search of hypotheses
Hypotheses in search of evidence
Evidentiary testing of hypotheses
Patient's complaints
New Tests
Medicine
Possible Charges or Complaints
Likeliness of Charges
Observations during Fact Investigation
New Potential Evidence
Law
Possible Causes
Likeliness of Causes
Observations at the Site of Incident
New Potential Evidence
Forensics
Possible Hypotheses or Explanations
Likeliness of New or Revised Theories
Observations of Events in Nature
New Observable Phenomena
Natural Science

Fig. 9. Evidence-based reasoning in various domains.

We therefore think that the developed computational theory of intelligence analysis could be extended into a general computational theory of evidence-based reasoning which could be the basis for developing learning agent shells for evidence-based reasoning in a variety of domains [17]. This would be a very significant extension of the applicability of the learning agent theory and technology discussed in this paper because, as Jeremy Betham stated over two centuries ago, "The field of evidence is no other than the field of knowledge" [18]. In fact, any agent that uses the information on the Internet needs to consider it as evidence rather than fact, and may use the type of reasoning discussed in this paper.

VII. Summary

We have reviewed an advanced learning-based approach to building practical cognitive assistants for intelligence analysis, where different types of experts had a staged contribution to the resulting agents. First scientists and knowledge engineers have formalized and taught a learning agent shell domain-independent knowledge for intelligence analysis. Then experts on various types of intelligence analysis problems have taught the resulting agent shell domain-specific knowledge for hypotheses analysis, with limited support from knowledge engineers. This resulted in specialized cognitive assistants for typical analysts who can analyze specific hypotheses based on evidence. During their use, the agents continue to learn reasoning patterns from their users. We have also shown that this approach could be generalized to developing cognitive assistants in other domains that involve evidence-based reasoning, such as medicine, law, science, forensics, history, anthropology, and others. Finally, this research suggests one approach by which non-computer scientists can develop their own cognitive assistants, not by developing them from scratch, but by further teaching already knowledgeable cognitive assistants, through very simple interactions.

References

[1] G. Tecuci, *Disciple: A Theory, Methodology and System for Learning Expert Knowledge*, Thése de Docteur en Science, University of Paris-South, July 1988.

[2] G. Tecuci, *Building Intelligent Agents: An Apprenticeship Multistrategy Learning Theory, Methodology, Tool and Case Studies*, San Diego: Academic Press, 1998.

[3] M. Boicu, *Modeling and Learning with Incomplete Knowledge*, PhD Thesis in Information Technology, Learning Agents Center, Volgenau School of Engineering, George Mason University, 2002.

[4] D. Marcu, *Learning of Mixed-Initiative Human-Computer Interaction Models*, Ph.D. Dissertation in Computer Science, Learning Agents Center, Volgenau School of Eng, George Mason University, 2009.

[5] G. Tecuci, M. Boicu, C. Boicu, D. Marcu, B. Stanescu, and M. Barbulescu, The Disciple-RKF Learning and Reasoning Agent, *Computational Intelligence, Vol.21, No.4*, pp. 462-479, 2005.

[6] G. Tecuci, M. Boicu, D. Marcu, C. Boicu, and M. Barbulescu, Disciple-LTA: Learning, Tutoring and Analytic Assistance, *Journal of Intelligence Community Research and Development*, July 2008.

[7] T. Simonite, "Bill Gates: Software Assistants Could Help Solve Global Problems," MIT Technology Review, 16 July 2013, http://www.technologyreview.com/news/517171/bill-gates-software-assistants-can-save-the-world/

[8] W3C, Semantic Web, http://www.w3.org/standards/semanticweb/

[9] D.A. Schum, *The Evidential Foundations of Probabilistic Reasoning*, Northwestern University Press, 1994, reprinting 2001.

[10] G. Tecuci, D. Marcu, M. Boicu, D.A. Schum, and K. Russell, Computational Theory and Cognitive Assistant for Intelligence Analysis, in *Proc. 6th Int Conf on Semantic Technologies for Intelligence, Defense, and Security*, pp. 68-75, Fairfax, VA, 2011.

[11] J. Cohen, *The Probable and the Provable*, Oxford, UK, Clarendon Press, 1977.

[12] L. Zadeh, The Role of Fuzzy Logic in the Management of Uncertainty in Expert Systems, *Fuzzy Sets and Systems*, Vol.11, pp. 199-227, 1983.

[13] M. Boicu G. Tecuci, and D.A. Schum, Intelligence Analysis Ontology for Cognitive Assistants, in *Proc. of the Conference "Ontology for the Intelligence Community: Towards Effective Exploitation and Integration of Intelligence Resources"* George Mason University, Fairfax, VA 2008.

[14] D.A. Schum, G. Tecuci, and M. Boicu, Analyzing Evidence and its Chain of Custody: A Mixed-Initiative Computational Approach, *Int. J. of Intelligence and Counterintelligence*, Vol.22, pp.298-319, 2009.

[15] G. Tecuci, M. Boicu, D. Marcu, D. Schum, and B. Hamilton, TIACRITIS System and Textbook: Learning Intelligence Analysis through Practice, in *Proc. of the 5th Int. Conference on Semantic Technologies for Intelligence, Defense, and Security* – STIDS 2010.

[16] G. Tecuci, D.A. Schum, M. Boicu, D. Marcu, *Introduction to Intelligence Analysis: A Hands-on Approach with TIACRITIS*, 220 pages, Learning Agents Center, George Mason University, 2010, 2011.

[17] G. Tecuci, D.A. Schum, M. Boicu, D. Marcu, K. Russell, Toward a Computational Theory of Evidence-based Reasoning, in *Proc. of the 18th International Conference on Control Systems and Computer Science*, University Politehnica of Bucharest, 24-27 May 2011.

[18] J. Betham, *An Introductory View of the Rationale of the Law of Evidence for Use by Non-lawyers as well as Lawyers* (vi *works* 1-187 (Bowring edition, 1837-43) originally edited by James Mill circa 1810.