AI Magazine Volume 28 Number 2 (2007) (© AAAI)

Seven Aspects of Mixed-Initiative Reasoning:

An Introduction to this Special Issue on Mixed-Initiative Assistants

Gheorghe Tecuci, Mihai Boicu, and Michael T. Cox

■ Mixed-initiative assistants are agents that interact seamlessly with humans to extend their problem-solving capabilities or provide new capabilities. Developing such agents requires the synergistic integration of many areas of AI, including knowledge representation, problem solving and planning, knowledge acquisition and learning, multiagent systems, discourse theory, and human-computer interaction. This paper introduces seven aspects of mixed-initiative reasoning (task, control, awareness, communication, personalization, architecture, and evaluation) and discusses them in the context of several state-of-the-art mixed-initiative assistants. The goal is to provide a framework for understanding and comparing existing mixed-initiative assistants and for developing general design principles and methods.

Mixed-initiative reasoning is concerned with the development of collaborative systems in which the human and automated agents work together to achieve a common goal in a way that exploits their complementary capabilities. Such systems can either accomplish goals unachievable by the component agents, assuming they work independently, or they can achieve the same goals more effectively. Mixed initiative assumes an efficient, natural interleaving of contributions by users and automated 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. Moreover, dynamic and flexible interaction facilitates adaptation to differences in knowledge, experience, and preferences among different users and to changes in the needs and preferences of individual users over time.

Mixed-initiative reasoning represents an important area of AI because of its potential of achieving both effective human-machine systems where humans interact seamlessly with agents, and multiagent systems whose capabilities are well above those of the component agents. This area has received considerable attention, as evidenced by a series of workshops[1, 2, 3, 4, 5] (Haller and McRoy 1997; Aha and Tecuci 2005).

A primary goal of this special issue is to present the current state of the art in the development and application of mixed-initiative assistants. To this purpose we have invited Eric Horvitz to share his thoughts on challenges and directions for research on mixed-initiative interaction (Horvitz 2007), and we have selected six representative papers. Three of the papers present general approaches to the development of mixed-initiative assistants (Ferguson and Allen 2007; Rich and Sidner 2007; Myers et al. 2007), one paper addresses the evaluation of a mixed-initiative planner (Cox and Zhang 2007), and two papers present successful applications of mixed-initiative assistants (Bresina and Morris 2007; Cheetham and Goebel 2007).

Eric Horvitz (2007) emphasizes the importance of the research on mixed-initiative interaction for understanding collaborative intelligence and for improving collaborative work, leading to new applications of automated reasoning, and enhancing our quality of life by changing how it feels to work with computers. He also identifies some of the great challenges and fascinating AI research opportunities for endowing computing systems with humanlike mixed-initiative interaction capabilities. These include seeking mutual understanding or grounding of joint activity, recognizing problem-solving opportunities, decomposing problems into subproblems, solving subproblems, combining solutions found by humans and machines, and maintaining natural communication and coordination during these processes.

Building on their prior work on the mixed-initiative dialogue and planning systems TRAINS (Ferguson, Allen, and Miller 1996) and TRIPS (Ferguson and Allen 1998), George Ferguson and James Allen (2007) present a practical, integrated approach to the design and implementation of a collaborative problem-solving assistant, further referred to here as TRIPS. The assistant integrates many capabilities required for collaboration, including reasoning, communication, planning, and execution. Its architecture includes a collaborative agent that is based on the belief-desire-intention (BDI) model of agency (Rao and Georgeff 1991) and a formal the-

 ISSN 0738-4602

ory of joint activity. Another key characteristic of the proposed approach is the use of representations for tasks that guide the assistant's collaborative behavior, allow it to interpret the behavior of others, and finally, allow it to deal with the shared beliefs and commitments that arise during collaboration.

Charles Rich and Candice Sidner (2007) present DiamondHelp, a generic collaborative task-guidance system, which can assist a user, for example, in programming a washing machine or a thermostat. DiamondHelp proposes a novel interface design for human-computer collaboration that combines an application-independent conversational interface adapted from online chat programs with application-specific direct-manipulation interfaces. This design preserves as much consistency as possible in the collaborative aspects of the interaction, so that different DiamondHelp applications have a similar look and feel. The DiamondHelp software can be used by others to easily construct such interfaces for new applications. Moreover, it can integrate the Collagen system (Rich, Sidner, and Lesh 2001) for representing SharedPlans (Grosz and Kraus 1996) of collaborators and modeling the dialogue state of the collaborators as they speak and perform activities. This not only further simplifies the use of DiamondHelp, but also provides it with more powerful collaboration capabilities.

Karen Myers, along with Pauline Berry, Jim Blythe, Ken Conley, Melinda Gervasio, Deborah McGuinness, David Morley, Avi Pfeffer, Martha Pollack, and Milind Tambe (2007) present PExA, a Project Execution Assistant that aids a busy knowledge worker by managing the user's time commitments (such as meetings and appointments) and by performing routine office tasks on the user's behalf. PExA integrates a diverse set of AI technologies within a BDI agent architecture. It provides a number of automated functions, but it is highly user centered in its support of human needs, responsiveness to human inputs, and adaptivity to user working style and preferences. Moreover, PExA illustrates several desirable qualities for a mixed-initiative assistant, including personalizability, directability, teachability, and transparency of operations.

Michael T. Cox and Chen Zhang (2007) argue that the traditional view of planning as search is not the correct metaphor to present to the user in a mixed-initiative interaction with an intelligent assistant. Instead the metaphor of planning as a goal-manipulation process is better suited to humans, especially the naïve users. In their GTrans interface to the Prodigy/Agent planning assistant, planning is cast as a process whereby the user minimally adapts the goals and resources associated with goals to compensate for limited resource availability or changes in the world state. The details of operator representations, variable bindings, and the underlying technology are hidden. To support this claim, they provide empirical results from an experimental study that evaluated groups of subjects using alternative software interfaces to the same underlying planning assistant. Given the goal-manipulation model, subjects tended to solve more goals with fewer steps than did subjects using an interface that presented a search-based planning methodology.

John L. Bresina and Paul H. Morris (2007) present MAPGEN, a successful mixed-initiative planner deployed as a mission-critical component of the ground operations system for the Mars Exploration Rover mission. It has been used daily for more than two years by the ground-planning personnel to collaboratively plan the activities of the *Spirit* and *Opportunity* rovers, with the objective of achieving as much science as possible while ensuring rover safety and keeping within the limitations of the rover's resources. MAPGEN provides a glimpse of how mixed-initiative assistants will change the nature of human problem solving. With the added efficiency resulting from the mixed-initiative approach, the human planners now have time to explore alternative "what-if" scenarios, perform solution fine-tuning that leads to a higher-quality plan, and are more willing to incorporate late-breaking information.

Finally, William Cheetham and Kai Goebel (2007) present STC, a mixed-initiative assistant that helps a call taker diagnose problems with home appliances. STC stores cases of problems and their solutions, a decision tree of questions that are used to differentiate the current case from all other cases, and rules that can automatically answer questions. STC is a successful implementation of a mixed-initiative assistant based on existing technology that both provides better service to customers and reduces the cost of this service. It has been in use since 1999 at multiple locations in the United States and has provided more than $50 million in financial benefits by increasing the percentage of questions that could be answered without sending a field service technician to the customers' homes.

Development of mixed-initiative assistants is very challenging because it requires the synergistic integration of many areas of AI, including knowledge representation, problem solving and planning, knowledge acquisition and learning, multiagent systems, discourse theory, and human-computer interaction. In order to better understand existing mixed-initiative systems and to help develop general design principles and methods for such systems, we have asked the authors to explicitly address in their papers how their systems deal with the issues of task, control, awareness, communication, personalization, architecture, and evaluation, as discussed in the following sections.

The Task Issue

The task issue regards the division of responsibility between the human and the agent for the tasks that need to be performed. In general, one develops a mixed-initiative assistant because there is some complementarity between a human and an automated agent with respect to the performance of particular tasks.

One dimension of complementarity between a human and an automated agent relates to their reasoning styles and computational strengths. Humans use common sense, intuition, creativity, and value systems in problem solving and decision making and can naturally interact with other humans. Automated agents do not have these capabilities, but they excel in speed of mathematical computations, can quickly store and retrieve large quantities of information, can effectively use deep and narrow subject matter expertise, and are not affected by stress or fatigue. For instance, in the STC system, the human call takers interact with the customers in natural language and the STC agent stores and retrieves the standardized knowledge about diagnosing appliances and guides the call taker in the diagnosis. In the case of MAPGEN, the user is responsible for higher-level planning decisions, such as which rover activities to plan next or which to unplan, while the agent generates the actual plan, ensuring plan validity with regard to mission flight rules or various temporal constraints. Moreover, because it is infeasible to formally encode and effectively utilize all the knowledge that characterizes plan quality, the user must also improve the plan generated by MAPGEN through manual fine-tuning.

When designing a human-agent mixed-initiative system one should assure that the operations to be performed by the human should be as natural and easy as possible. For instance, Cox and Zhang (2007) analyze two ways in which a human can guide the planning performed by an assistant. In one case the user chooses search-specific decision alternatives, while in the other she chooses goal alternatives to the problem specification. The second operation is more natural to the user and more likely to lead to better overall plans, as confirmed by the provided experimental results.

Another dimension of complementarity between a human and an automated agent relates to their relative expertise with respect to the tasks to be performed. At one extreme, an expert assistant can guide a novice user in performing some tasks, as illustrated by DiamondHelp. At the other extreme, an expert user can focus on strategic problem solving and delegate routine tasks to the agent, as illustrated by PExA. For instance, PExA relieves the user of the responsibility for such frequently occurring and routine tasks as meeting scheduling or expense reimbursement. In between these extremes are the situations when the expertise is distributed between the human and the agent, and the two have to collaborate to achieve a common goal. This is illustrated by MAPGEN, which produces generic plans, the quality of which are improved by the human planners through fine-tuning.

In general, who does what is a matter of agreeing, through dialogue, on the allocation of task responsibility and then jointly committing to the successful performance of the tasks. However, many systems are designed with a certain expected division of responsibility. For instance, DiamondHelp assumes that the user knows what she wants to do at a high level but needs help carrying out the necessary details.

The division of the tasks between the human and the assistant does not need to be fixed. For instance, a key design characteristic of MAPGEN is to assure a user-adjustable level of autonomy of the planning assistant. At the full-automation end of the spectrum, the assistant generates a complete plan by itself. At the other extreme, the user can manually insert an activity in a plan. In between, the user may ask the assistant to insert an activity anywhere into the current partial plan such that all constraints are satisfied.

An important design decision of TRIPS is to keep task specifications separate from the capabilities of the agents who perform them, allowing the tasks to be performed by different combinations of agents under different conditions. In such a case, the division of responsibility can be dynamic and flexible, able to be discussed and renegotiated at any time. TRIPS also illustrates some general features that the task representation language for mixed-initiative systems must allow, such as: ability to represent partial knowledge, ability to represent knowledge requirements for a task, ability to represent tasks at different levels of abstraction, ability to represent execution of tasks by agents and also to support a natural communication through task description and explanation.

The Control Issue

The control issue regards the strategies for shifting the initiative and control between the human and the agent, including proactive behavior. Deciding who should do what and when is a complex problem that depends not only on the qualifications of the participants but also on the set of tasks that need to be performed at a certain moment.

In principle, the human and the agent should be in control of those tasks that optimize some global measure of their joint performance. However, this is difficult to assess and may result in conflicts when each participant believes that it should be in control. A way to resolve such conflicts and, in general, to shift the initiative, is through interaction. TRIPS accomplishes this in a collaboration framework based on the BDI model of agency where the human and the agent operate continuously, asynchronously and in parallel, based on joint commitments. Communicative initiative is driven by the agent's need of knowledge. This framework allows continuous interpretation of user action and input, interleaved and overlapping generation of agent's output, and independent actions by the agent in pursuit of its own desires and goals.

PExA also relies on a BDI model of agency, but specializes it to a delegative interaction in which the user decides what needs to be done and which tasks he or she feels comfortable allocating to the agent. Then the agent operates in a fairly autonomous manner, interacting with the user to solicit necessary information and to confirm important decisions. The agent also manifests proactive behavior to inform the user of problems, to provide reminders of user commitments, and to provide feedback on user requests.

In DiamondHelp (with Collagen) control is managed by maintaining a discourse state comprising a focus stack and goal decomposition tree and updating it based on the occurring events and the task model. Based on these, a prioritized list of actions is produced from which the agent may select the next action.

Cheetham and Goebel (2007) proposed an even simpler mechanism of control in which the actions of the agent are sorted by the confidence that the initiative should be taken and the best action is executed. However, accurately computing such confidence factors remains a challenge for complex applications.

Horvitz (1999) identified several deficiencies of the current automated agents that support a user, such as poor guessing about the user's goals and needs, inadequate consideration of the costs and benefits of their actions, poor timing of the actions, and inadequate attention to opportunities that would allow the user to guide the invocation of the agents to refine their results. In response, he proposed the following set of design principles, many of them with direct impact on the control issue: (1) developing significant value-added automation; (2) considering uncertainty about a user's goals; (3) considering the status of a user's attention in the timing of services; (4) inferring ideal action in light of costs, benefits, and uncertainties; (5) employing dialogue to resolve key uncertainties; (6) allowing efficient direct invocation and termination; (7) minimizing the cost of poor guesses about action and timing; (8) scoping precision of service to match uncertainty, variation in goals; (9) providing mechanisms for efficient agent-user collaboration to refine results; (10) employing socially appropriate behaviors for agent-user interaction; (11) maintaining working memory of recent interactions; and (12) continuing to learn by observing.

MAPGEN illustrates the usefulness of some of these principles. For instance, an earlier version of MAPGEN was continuously and aggressively taking initiative to ensure the validity of the generated rover mission plan with respect to various factors, such as science constraints or mission flight rules. If the user performed operations that would produce an inconsistency, such operations were immediately undone by MAPGEN. This type of initiative was regarded as a little too aggressive by the users, who wanted to have the option to (at least temporarily) violate a flight rule or science constraint. As a result, the constraint-enforcement facility of MAPGEN was redesigned to be more passive and user adjustable. For instance, MAPGEN now constantly performs passive violation checking but applies active enforcement of constraints only when the user requests it.

As another example of applying some of the above principles, the STC system automatically answers some questions to help in diagnosis, but answering them does not interrupt the user. Instead, the call taker can, at any time, change an automatically generated answer.

The Awareness Issue

The awareness issue regards the maintenance of a shared understanding of the evolving state of the problem-solving process by the human and the agent. In essence the collaborating agents need to share basic facts and beliefs and to have a common understanding of their joint goals, a transparent reasoning process, and a common understanding of the results. This is crucial for effective human-agent mixed-initiative reasoning, but it is difficult to achieve because humans and automated agents have completely different interaction modalities and understanding capabilities.

Maintaining shared awareness is the guiding principle of the TRIPS family of mixed-initiative systems. Communication and dialogue is used both to reach agreement on facts, beliefs, and goals and later to update, maintain, and exploit a shared state of knowledge for effective problem solving.

DiamondHelp relies on the combination of the application-specific direct-manipulation interface and the generic chat window and scroll bar to maintain shared awareness of the problem-solving process. If Collagen is incorporated into DiamondHelp, it can provide a more complete representation of the task and conversation state in the form of a segmented interaction history.

For STC, in order for the agent to be able to make valid appliance diagnostic suggestions, it needs to have awareness of all the information that the call taker has about the problem. The call taker must also have awareness about what the agent is doing. Because the agent can take the initiative to answer questions, the user must be able to inspect the conclusions that the agent has made.

Transparency is an essential component of shared awareness. To accept the agent's assistance, the user needs to have a clear understanding of the agent's actions, reasoning, and conclusions. PExA leverages inference web explanation infrastructure (McGuinness and Pinheiro da Silva 2004) and, for instance, uses several context-dependent strategies to answer a variety of questions, including why it is currently performing a task, why the task is not yet finished, what information it relies on, and how it will execute something. One of its interesting capabilities is that of generating possible context-appropriate follow-up questions for the user to ask (for example, requests for additional detail, clarifying questions about an explanation that has been provided previously, or questions requesting that an alternate strategy be used for answering a previously posed question).

In the case of STC, call takers and customers often wonder why the system is suggesting a specific question. User trust is enhanced if there is a clear explanation for why the system is taking some action. When the questions were defined by the system developers, they also created explanations for why the questions are asked. These explanations can be displayed for the call taker by clicking on the questions in the user interface.

For some types of problems, transparency may be quite difficult to achieve because of the complexity of the reasoning process and of the generated solution. For instance, MAPGEN generates a family of complex plans (each with up to 100 top-level activities and 3500 lower-level activities) with a range of start times, but it can only display a grounded plan with fixed start times. Additionally, the user is largely unaware of the ordering constraints that the planner has imposed in order to satisfy mutual-exclusion flight rules. All these make the process of fine-tuning the plan by the user more complicated. Dealing with such cases requires the development of methods for generating abstract but clear explanations that do not overwhelm the user with a myriad of unimportant details.

The Communication Issue

The communication issue regards the protocols that facilitate the exchange of knowledge and information between the human and the agent, including mixed-initiative dialogue and multimodal interfaces. In principle, the human-agent communication needs to be as natural and efficient as possible for the human, and as complete and unambiguous as possible for the agent, but these are often competing goals.

Ferguson and Allen (2007) promote the use of spoken natural language dialogue because (1) this is a very efficient means of communication for people; (2) it requires little or no training to use; (3) it gives the greatest insight into the nature of human communication and collaboration; and (4) it is the most likely way to achieve true mixed-initiative, collaborative systems. They formulate two main requirements for a general interface (whether graphical or natural language based): to support interpretation, the context displayed or implied by the interface must be made explicit and available for use by the agent's interpretation and collaboration components; and the actions permitted by the interface must be expressed in terms of communicative acts with semantically meaningful content.

DiamondHelp uses the scrolling speech bubble metaphor inspired by the online chat programs for human-human communication to enable the conversation between the human and the agent. The system exploits the characteristics of its application domain (guiding the human to use a device) to implement a flexible protocol combining chatlike conversation with direct manipulation that gives the feeling of natural communication, without actually requiring natural language or speech processing.

One approach to avoid or at least limit the complexities of natural language processing is to use a communication protocol that takes into account two complementary capabilities of humans and agents. First, that it is easier for a human to understand sentences in the formal language of the agent than it is to produce such formal sentences. Second, that it is easier for the agent to generate formal sentences than it is to understand sentences in the natural language used by the human. This approach was very successfully used in the Disciple system (Tecuci 1998; Boicu, Tecuci, and Marcu 2005) for the acquisition of problem-solving knowledge directly from subject matter experts. Instead of asking the expert to provide an explanation of why a problem-solving episode is correct, Disciple proposes a list of plausible explanations, asking the expert to choose the correct one. A similar idea is also used in PExA where the user provides an informal textual description of a task to be performed by the agent and the agent responds with a list of possible tasks for the user to choose from.

GTrans illustrates a novel communication mode where the human can modify the goals of the planning system. For example if the goal is to make a river impassable, and not enough air units exist to destroy all bridges across the river, the user can change the goal to limit the transportation capacity over the river. The GTrans system supports communication of intent through various changes or transformations on goal predicates. The interface interacts with the user through pull-down menus and interactive activities that keep the reasoning focused upon what the user wants to achieve rather than the technical details related to specific planning algorithms.

Finally, in order to simplify the interaction with the user, both DiamondHelp and PExA promote the use of a uniform interface for all the components and applications of the system. Thus, for instance, if the user is familiar with one DiamondHelp application, she should know how to use any other DiamondHelp application.

The Personalization Issue

The personalization issue regards the adaptation of the agent's knowledge and behavior to its user's problem-solving strategies, preferences, biases, and assumptions. Personalization is also crucial to effective collaboration, both enabling the system to more quickly produce solutions that are likely to be acceptable or desirable to the user and helping the user to avoid mistakes by checking her biases and assumptions.

DiamondHelp employs two simple but effective personalization mechanisms that take advantage of Collagen's capabilities, such as its use of a student model. The implicit control strategy in DiamondHelp is to return control to the user as quickly as possible. However, based on simple observations of the user's behavior, such as timing and errors, it can switch into a mode in which it takes control and guides the user through the execution of an entire task. A second personalization has to do with whether the agent asks the user to perform certain manipulations on the application GUI or simply performs them itself. DiamondHelp can switch between these modes, depending on whether the user has already performed the current action once or twice herself.

Personalization is the main goal of PExA. This is achieved through a combination of explicitly stated user preferences and active learning. First the user specifies her initial preferences and their relative trade-offs through a graphical tool, from which PExA induces an initial multicriteria evaluation function. This function is further improved through active learning that captures the user's unstated or evolving preferences.

One natural way to personalize the agent is for the user to teach it directly how to solve problems. Disciple (Tecuci et al. 2005), for instance, uses methods of mixed-initiative problem solving, integrated teaching and learning, and multistrategy learning to enable a subject matter expert to teach it in a way that resembles how the expert would teach a person. The expert provides examples on how to solve specific problems, helps Disciple to understand the solutions, and supervises and corrects its problem-solving behavior. Disciple learns from the expert by generalizing the examples and building and refining its knowledge base. In essence, this creates a synergism between the expert that has the knowledge to be formalized and the agent that knows how to formalize it but also results in a highly specialized agent that behaves as an extension of the problem-solving capabilities of the expert.

PExA can also be trained by its user who can directly change its behavior by adding new steps in a procedure, modifying conditions, and changing step orderings, without needing to have knowledge of PExA's procedure representation or precise domain ontology. PExA also keeps track of the modifications and can later explain why it is behaving the way it is (as the result of a modified procedure) and can explain how, when, and by whom the modification was done.

Thus teachability is an important desired capability of a mixed-initiative assistant, and not only because it allows a natural personalization of the agent, but also because it allows the combined human-agent system to adapt easier to changes in the application domain.

The Architecture Issue

The architecture issue regards the design principles, methodologies, and technologies for different types of mixed-initiative roles and behaviors. Identifying and studying them will significantly facilitate the development of useful mixed-initiative systems and will lead to a wider applicability and acceptance of artificial intelligence.

The systems described in this special issue illustrate some good architectural practices. One is to separate the communication from control, as in TRIPS, which includes three main agents: the interpretation agent that interprets the user's actions, the generation agent that generates the output to the user, and the collaborative agent. The collaborative agent interacts with the other components through collaborative problem-solving acts, independent of the actual communication modality adopted (be it spoken or written natural language or a graphical interface). Yet another architectural practice emerging from TRIPS is to represent and reason with the system's core competencies as tasks at the metalevel, allowing the modification and improvement of the various aspects of system performance.

A third good architectural practice used both by TRIPS and by PExA is to assure asynchronous behavior of the agents in their multiagent systems. Fourth, DiamondHelp's software architecture of reusable JavaBeans is a good illustration of component reuse. Finally, fifth, both DiamondHelp and PExA promote the employment of a uniform interface across their many components to facilitate the system's use.

The Evaluation Issue

The evaluation issue is related to the human and automated agent contribution to the emergent behavior of the system and the overall system's performance versus fully automated, fully manual, or alternative mixed-initiative approaches.

In spite of its importance, with few exceptions (Oates and Cohen 1994; Guinn 1998; Cortelessa and Cesta 2005; Kirkpatrick, Dilkina, and Havens 2005), not much work has been done to define evaluation frameworks for mixed-initiative systems or to conduct significant experiments to differentiate empirically the relative contributions to performance. This is partly due to the following factors: (a) the mixed-initiative systems are generally very complex, with components for reasoning, communication, planning, execution, and learning, and therefore difficult to evaluate; (b) the evaluation has to involve different types of users and is therefore very costly and time-consuming, (c) the evaluation requires several comparisons, with fully automated, fully manual solutions, and alternative mixed-initiative approaches.

Cox and Zhang (2007) evaluate some aspects of mixed-initiative planning systems. They have held constant the contribution of the intelligent agent and varied the model of the cognitive process presented to the human user at the software interface. In one group, planning was presented as a search process, whereas in a second group, planning was presented as goal manipulation. Given these two conditions they have shown a differential effect on performance, although the awareness issue differed across each condition. What was not examined, however, was the relative effect on performance given different task distributions, for example.

Ferguson and Allen (2007) emphasize the use of end-to-end or task-based measures of system performance, as opposed to component measures, because poor performance by any given component might be compensated for by another, and stellar performance by a single component is not guaranteed to translate into user satisfaction.

Rich and Sidner (2007) outline three conditions in a user study planned for the evaluation of DiamondHelp using the washer-dryer case. In each condition the users will be assigned the same set of tasks requiring the use of the advanced programmability features of the washer-dryer. In condition A, the users will have no guidance and no access to user manuals. In condition B, the users will have access to a printed manual that contains literally the same text that is communicated dynamically by DiamondHelp in condition C. They plan to obtain both objective measures, such as time and quality of task completion, and subjective evaluations of experience.

Conclusion

Humans have limitations that intelligent agents may alleviate, allowing us to cope better with the increasing challenges of the information and knowledge society. This requires that intelligent agents become essential components of our future systems and organizations. In fact, our future computers and most of the other systems and tools will gradually become intelligent agents.

The main goal of the research on mixed-initiative assistants is to lead to the development of agents that are easy to use and are truly helpful. These agents should represent significant extensions of our capabilities or provide us with new capabilities that we can employ in a natural way.

Because of the complexity involved in developing mixed-initiative assistants, we have isolated seven issues (task, control, awareness, communication, personalization, architecture, and evaluation) that help not only to understand and compare existing mixed-initiative assistants but also to develop general design principles and methods for such systems. These mixed-initiative issues are not independent and interact in complex ways, as illustrated by each system described in the follow-on papers.

Acknowledgements

This work was partially supported by the Air Force Research Laboratory under agreement number FA8750-04-1-0527 and by the Air Force Office of Scientific Research under agreement number FA9550-07-1-0268. 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, or the U.S. Government.

Notes

1. See ECCBR'02: Workshop on Mixed-Initiative Case-Based Reasoning, held during the Sixth European Conference on Case-Based Reasoning, 4 September, Robert Gordon University, Aberdeen, Scotland, UK (home.earthlink.net/~dwaha/research/meetings/eccbr02-micbrw/).

2. See CfP: Workshop on Mixed-Initiative Case-Based Reasoning, held during the Fifth International Conference on Case-Based Reasoning, 24 June 2003, Norwegian University of Science and Technology, Trondheim, Norway. (home.earthlink.net/~dwaha/ research/meetings/iccbr03-micbrw).

3. AAAI-99 Workshop on Mixed Initiative Intelligence, held 19 July 1999, Omni Rosen Hotel, Orlando, FL, Michael T. Cox, chair (www.mcox.org/mii).

4. See the ICAPS 2005 Workshop on Mixed-Initiative Planning and Scheduling, held in conjunction with the Fifteenth International Conference on Automated Planning and Scheduling, Monterey, California, 5 June, organized by G. Ferguson, C. Haye, and G. Sullivan (www.cs.rochester.edu/research/mipas2005).

5. See the 2003 IJCAI Workshop on Mixed-Initiative Intelligent Systems, organized by G. Tecuci, D. Aha, M. Boicu, M. Cox, G. Ferguson, and A. Tate, held 9 August in Acapulco, Mexico (lac.gmu.edu/MIIS/default.htm).

References

Aha, D., and Tecuci, G., eds. 2005. Mixed-Initiative Problem-Solving Assistants: Papers from the 2005 AAAI Fall Symposium. Technical Report FS-05-07, Association for the Advancement of Artificial Intelligence, Menlo Park, CA.

Boicu, M.; Tecuci, G.; Marcu, D. 2005. Mixed-Initiative Assistant for Modeling Expert's Reasoning. In Mixed-Initiative Problem-Solving Assistants: Papers from the 2005 AAAI Fall Symposium, ed. D. Aha and G. Tecuci. Technical Report FS-05-07, Association for the Advancement of Artificial Intelligence, Menlo Park, CA.

Bresina, J. L.; and Morris, P. H. 2007. Mixed-Initiative Planning in Space Mission Operations. *AI Magazine* 28(2).

Cheetham, W., and Goebel, K. 2007. Appliance Call Center: A Successful Mixed-Initiative Case Study. *AI Magazine* 28(2).

Cortellessa, G., and Cesta, A. 2005. Towards a Reliable Evaluation of Mixed-Initiative Systems. In Mixed-Initiative Problem-Solving Assistants: Papers from the 2005 AAAI Fall Symposium, ed. D. Aha and G. Tecuci. Technical Report FS-05-07, Association for the Advancement of Artificial Intelligence, Menlo Park, CA.

Cox, M. T., and Zhang, C. 2007. Mixed-Initiative Goal Manipulation. *AI Magazine* 28(2).

Ferguson, G., and Allen, J. 2007. Mixed-Initiative Systems for Collaborative Problem Solving. *AI Magazine* 28(2).

Ferguson, G.; Allen, J.; and Miller, B. 1996. TRAINS-95: Towards a Mixed-Initiative Planning Assistant. In *Proceedings of the Third Conference on Artificial Intelligence Planning Systems* (AIPS-96), 70–77. Menlo Park, CA: AAAI Press.

Ferguson, G., and Allen, J. 1998. TRIPS: An Intelligent Integrated Problem-Solving Assistant. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence* (AAAI-98), 567–573. Menlo Park, CA: AAAI Press.

Grosz, B. J., and Kraus, S. 1996. Collaborative Plans for Complex Group Action. *Artificial Intelligence* 86(2): 269–357.

Guinn, C. I. 1998. An Analysis of Initiative Selection in Collaborative Task-Oriented Discourse. *User Modeling and User-Adapted Interaction* 8(3): 255–314.

Haller, S., and McRoy, S., eds. 1997. Computational Models for Mixed Initiative Interaction: Papers from the 1997 AAAI Spring Symposium. Technical Report SS-97-04, Association for the Advancement of Artificial Intelligence, Menlo Park, CA.

Horvitz, E. 1999. Principles of Mixed-Initiative User Interfaces. In *Proceedings of the 1999 ACM SIGCHI Conference on Human Factors in Computing Systems,* 159–166. New York: ACM Press.

Horvitz, E. 2007. Reflections on Challenges and Promises of Mixed-Initiative Interaction. *AI Magazine* 28(2).

Kirkpatrick, A.; Dilkina, B.; and Havens, W. 2005. A Framework for Designing and Evaluating Mixed-Initiative Optimization Systems. Paper presented at the ICAPS Workshop on Mixed-Initiative Planning and Scheduling, held in conjunction with the Fifteenth International Conference on Automated Planning and Scheduling, Monterey, California, June.

McGuinness, D. L., and Pinheiro da Silva, P. 2004. Explaining Answers from the Semantic Web: The Inference Web Approach. *Journal of Web Semantics* 1(4).

Myers, K.; Berry, P.; Blythe, J.; Conley, K.; Gervasio, M.; McGuinness, D.; Morley, D.; Pfeffer, A.; Pollack, M.; and Tambe, M. 2007. An Intelligent Personal Assistant for Task and Time Management. *AI Magazine* 28(2).

Oates, T., and Cohen, P. R. 1994. Mixed-initiative Scheduling Maintenance: A First Step toward Plan Steering. In *ARPA/Rome Laboratory Knowledge-Based Planning and Scheduling Initiative: Workshop Proceedings*, ed. M. Burstein, 133–143. San Francisco: Morgan Kaufmann.

Rao, A., and Georgeff, M. 1991. Modeling Rational Agents within a BDI-Architecture. In *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning (KR-91),* ed. J. Allen, R. Fikes, and E. Sandewall, 473–484. San Francisco: Morgan Kaufmann Publishers.

Rich, C.; Sidner, C. L.; and Lesh, N. 2001. Collagen: Applying Collaborative Discourse Theory to Human-Computer Interaction. *AI Magazine* 22(4): 15–25.

Rich, C.; and Sidner C. L. 2007. DiamondHelp: A Generic Collaborative Task Guidance System. *AI Magazine* 28(2).

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

Tecuci, G.; Boicu, M.; Boicu, C.; Marcu, D.; Stanescu, B.; Barbulescu, M. 2005. The Disciple-RKF Learning and Reasoning Agent. *Computational Intelligence* 21(4): 462–479.

Gheorghe Tecuci is a professor of computer science in the Volgenau School of Information Technology and Engineering and Director of the Learning Agents Center (lac.gmu.edu) at George Mason University. Tecuci is also a member of the Romanian Academy and former chair of artificial intelligence at the U.S. Army War College. He received two Ph.D.'s in computer science, from the University of Paris-South and from the Polytechnic University of Bucharest. Tecuci has published more than 150 papers including the book *Building Intelligent Agents: An Apprenticeship Multistrategy Learning Theory, Methodology, Tool, and Case Studies*. His research focuses on developing and applying a general theory of how subject matter experts can directly teach automated agents how to solve problems. This will enable noncomputer scientists to be not only users of programs developed by others, as they are today, but also agent developers themselves. His e-mail address is tecuci@gmu.edu.

Mihai Boicu is a research assistant professor and associate director of the Learning Agents Center in the Volgenau School of Information Technology and Engineering of 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. His domains of interest are knowledge representation, knowledge acquisition, multistrategy learning, and mixed-initiative reasoning with applications to instructable agents. Boicu has published around 50 papers in these areas and has received several awards and recognitions for his professional activity, including the outstanding graduate student award from George Mason University, the deployed application award from the American Association of Artificial Intelligence, and two certificates of appreciation and the centennial coin from the U.S. Army War College. His e-mail address is mboicu@gmu.edu.

Michael T. Cox is a senior scientist in the Intelligent Distributing Computing Department of BBN Technologies, Cambridge, Massachusetts. Previous to this position, Cox was an assistant professor in the Department of Computer Science and Engineering at Wright State University, Dayton, Ohio, where he was the director of Wright State's Collaboration and Cognition Laboratory. His research interests include case-based reasoning, collaborative mixed-initiative planning, understanding (situation assessment), introspection, and learning. More specifically, he is interested in how goals interact with and influence these broader cognitive processes. His e-mail address is mcox@bbn.com.