AI Magazine Volume 27 Number 2 (2006) (© AAAI)

Companion Cognitive Systems
A Step toward Human-Level AI

Kenneth D. Forbus and Thomas R. Hinrichs

■ We are developing Companion Cognitive Systems, a new kind of software that can be effectively treated as a collaborator. Aside from their potential utility, we believe this effort is important because it focuses on three key problems that must be solved to achieve human-level AI: Robust reasoning and learning, interactivity, and longevity. We describe the ideas we are using to develop the first architecture for Companions: analogical processing, grounded in cognitive science for reasoning and learning, sketching and concept maps to improve interactivity, and a distributed agent architecture hosted on a cluster to achieve performance and longevity. We outline some results on learning by accumulating examples derived from our first experimental version.

We believe that this is an exciting time for research in artificial intelligence and cognitive science. What is known about cognition has grown significantly, and communication between fields in cognitive science has catalyzed all of them. Large-scale representational resources, such as WordNet, FrameNet, and Cyc, have become available so that one can build large knowledge systems without starting from scratch. Central processing units (CPUs) have become fast enough and memories large enough to tackle systems that could only be dreamed about previously. The confluence of these three factors suggest to us that the time is right for more ambitious projects, building integrated systems using the best available results from cognitive science.

The effort we have embarked on to create *Companion Cognitive Systems* represents one such project. Let us start with our practical goals for Companions. The problems we face are growing more complex, but we are not becoming any smarter. Software can help, but often it becomes part of the problem by adding new layers of complexity. We need to bring software closer to us, improving conceptual bandwidth and having it adapt to us, rather than the other way around. Our vision is this: Companions will be software aide-de-camps, collaborators with their users. Companions will help their users work through complex arguments, automatically retrieving relevant precedents, providing cautions and counter-indications as well as supporting evidence. Companions will be capable of effective operation for weeks and months at a time, assimilating new information, generating and maintaining scenarios and predictions. Companions will continually adapt and learn, about the domains they are working in, their users, and themselves.

It is useful to distinguish the Companions vision from the Defense Advanced Research Projects Agency (DARPA) Perceptive Agents that Learn (PAL) Program. While similar in many respects, our ambitions are somewhat more modest. PALs are intended to be like Radar O'Riley: socially aware, for example, and capable of participating in multiperson spoken dialogues. Companions are intended to be, for lack of a better term, nerd sidekicks—assistants who are heavily focused on a couple of areas of interest to you, and nothing else. This lets us factor out several problems, including natural language understanding, low-level vision, and robotics. Companions are, in essence, the classic "mind in a box" model of an artificial intelligence.

Even within these limitations, creating Companions is forcing us to tackle several problems that are crucial for building human-level AIs:

Robust reasoning and learning. Companions will have to learn about their domains, their users, and themselves.

Longevity. Companions will need to operate continuously over weeks and months at a time.

Interactivity. Companions must be capable of high-bandwidth interaction with their human partners. This includes being capable of taking advice.

Our Approach

The following are the ideas we are using to create Companions:

Robust Reasoning and Learning

Our working hypothesis is that the flexibility and breadth of human commonsense reasoning and learning arises from analogical reasoning and learning from experience (Forbus and Gentner 1997). Within-domain analogies provide rapid, robust predictions. Analogies between domains can yield deep new insights and facilitate learning from instruction. First-principles reasoning emerges slowly, as generalizations created from examples incrementally through analogical comparisons. This hypothesis suggests a very different approach to building robust cognitive software than is typically proposed. Reasoning and learning by analogy are central, rather than exotic operations undertaken only rarely. Accumulating and refining examples becomes central to building systems that can learn and adapt. Our cognitive simulations of analogical processing (SME for analogical matching [Falkenhainer, Forbus, and Gentner 1989; Forbus, Ferguson, and Gentner 1994], MAC/FAC for similarity-based retrieval [Forbus, Gentner, and Law 1995], and SEQL for generalization [Kuehne, Genter, and Forbus 2000]) form the core components for learning and reasoning. These components are based on Gentner's structure-mapping theory of analogy and similarity (see, for example, Gentner [1983]), a psychological account with considerable evidence behind it (compare with Gentner and Markman 1997).

These simulations have been successfully used to model a large number of psychological findings, and both SME and SEQL have been used to make new psychological predictions which, in turn, have been borne out in experiments with human subjects (for example, Kuehne, Forbus, and Gentner [2000]). SME and MAC/FAC have already been used in performance systems, involving several different large knowledge bases (for example, Forbus, Mostek, and Ferguson [2002]; Mostek, Forbus, and Meverden [2000]). One of the key hypotheses we are testing in Companions is the idea that most learning and reasoning can be handled through analogical processing. In other words, it's structure-mapping all the way down.

Longevity and Performance

Companions will require a combination of intense interaction, deep reasoning, and continuous learning. We plan to achieve this by using a distributed agent architecture, hosted on cluster computers, to provide task-level parallelism. The particular distributed agent architecture we are using evolved from our RoboTA distributed coaching system (Forbus and Kuehne 1998), which uses KQML (Labrou and Finin 1997) as a communication medium between agents. A Companion is made up of a collection of agents, spread across the CPUs of a cluster. We are assuming ultimately at least 10 CPUs per Companion, so that, for instance, analogical retrieval of relevant precedents proceeds entirely in parallel with other reasoning processes, such as the high-level visual processing involved in understanding a user's sketched input.

Robustness will be enhanced by making the agents "hot-swappable," that is, the logs maintained by the agents in operation will enable another copy to pick up (at a very coarse granularity) where a previous copy left off. This will enable an agent whose memory is clogging up (or crashes) to be taken offline, so that its results can be assimilated while another agent carries on with the task. This scheme requires replicating the knowledge base and case libraries as necessary to minimize communication overhead, and broadcasting working memory state incrementally, using a publish/subscribe model, as well as disk logging. These logs will also be used for adaptation and knowledge reformulation. Just as a dolphin only sleeps with half of its brain at a time, our Companions will use several CPUs to test proposed changes by "rehearsing" them with logged activities, to evaluate the quality and performance payoffs of proposed learned knowledge and skills.

Interactivity

People communicate with each other in a variety of ways. Natural language is certainly one method, but we also sketch, and people who communicate with each other frequently evolve their own shared signals for communication. We are not tackling natural language in this project, since by itself it harbors several equally ambitious projects in it. Instead, we are building on our work on sketch understanding (Forbus and Usher 2002; Forbus, Usher, and Chapman 2003) to provide high bandwidth interaction. Our nuSketch approach is uniquely suited for Companions, since it does not restrict us to narrow domains.

Not everything can be sketched, of course. For other kinds of interactions, we are building on the idea of concept maps (Novak and Gowin 1984). Our VModel qualitative modeling system, which uses concept maps founded on semantics from qualitative process (QP) theory, has been successfully used by middle-school students in classroom experiments in the Chicago Public School system (Forbus et al. 2004). This experience, along with SRI's SHAKEN system (Thomere et al. 2002) and the UWF work on concept maps (Cañas et al. 1995), leads us to believe that we can make concept map systems that professionals will find useful. Specifically, we have developed a new kind of concept map, *relational concept maps*. A relational concept map supports statements involving n-ary relationships, instead of just binary relationships as in traditional concept maps. That is, statement nodes are themselves reified, and the only links in a relational concept map are argument links, tying the node for a statement to the nodes representing its arguments. This increased expressive power provides a useful complement to our sketching interfaces.

Modeling in Companions

We expect that several kinds of models will play important integrative roles in Companions. Models will be used to mediate and coordinate between interaction modalities, summarize internal state, and facilitate user interaction. For a Companion to provide understandable explanations and persuasive arguments or to behave robustly, we believe it must have concise, accessible representations of its current state, beliefs about the user, and dynamic context. We see four kinds of specialized models as being crucial: (1) *Situation* and *domain models* capture the current problem and relevant knowledge about it. (2) *Task* and *dialogue models* describe the shared task and where the human/computer partnership is in working on it. (3) *User models* capture the idiosyncratic preferences, habits, and utilities of the human partner(s). (4) *Self-models* provide the Companion's own understanding of its operations, abilities, and preferences (Cox 2005, McDermott 2001). Exploring how systems can learn, adapt, and exploit these models over the long term to support interaction and problem solving is one of the major goals of this project. We discuss each kind of model briefly in turn.

Situation and Domain Models. Our experimental domains, described later, leverage our earlier research on qualitative process theory (Forbus 1984), qualitative mechanics (Kim 1997, Nielsen 1988), compositional modeling (Erol, Nau, and Hendler 1994), and sketch understanding. We are also drawing upon a subset of the Cyc KB contents, including much of the knowledge associated with the upper ontology and the DARPA-generated materials about military topics and international relations generated in the HPKB, CPOF, and RKF programs. This knowledge forms the starting endowment for Companions.

Task and Dialogue Models. Our task models are expressed as HTN's (Erol, Nau, and Hendler 1994). For dialogue models, we are using script representations from ResearchCyc, which are being extended with ideas from Allen, Ferguson, and Stent (2001). We are developing new models for knowledge capture and joint problem-solving dialogues, including drill-down and advice-taking.

User Models. By keeping track of decisions and choices made by the user(s) in the logs, a statistical model of user preferences and conventions will be built up (Horvitz et al 2003 2004). Some of this information will clearly be about a specific user, such as preferred order of tackling subproblems. However, some of it will be more widely shared conventions (such as when drawing a wheelbarrow, the glyph representing the axle of the wheel provides a connection that leaves the wheel free to rotate), as we discuss later on.

As a user interacts with a Companion, at least one agent will be watching the interaction with standing goals to acquire knowledge about the user's intent, satisfaction with results, and abstract patterns of interaction. It will be fleshing out candidate task and preference models to learn, for example, that when

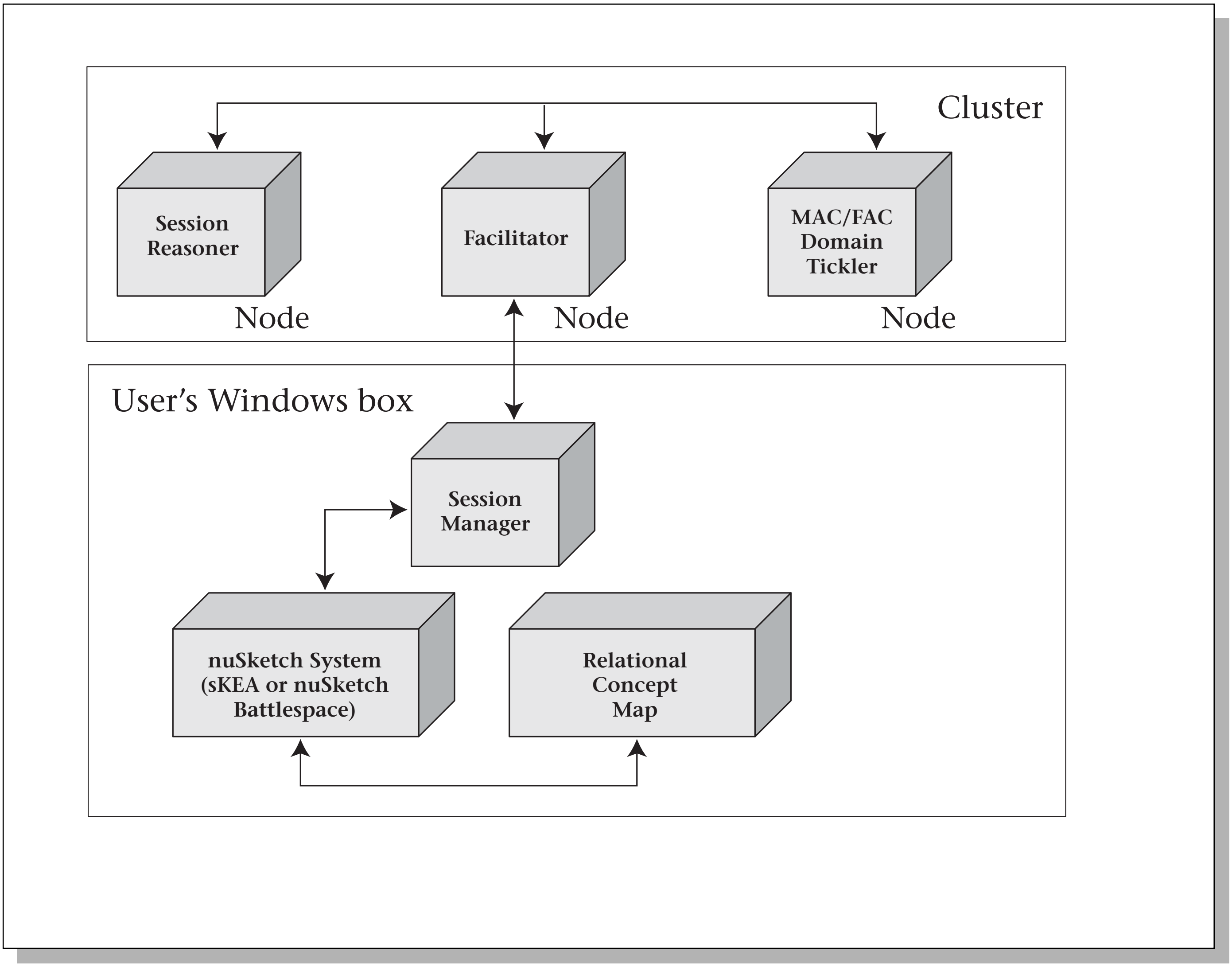

Figure 1. First Operational Companion Architecture.

the user is characterizing a situation, they typically want both positive and negative precedents, though perhaps not simultaneously.

Self Models. Our starting point for generating self-models are the machine-readable logs of user interactions and internal operations, described previously. In addition to providing checkpoints/rollback for KB contents and hot-swapping, they will serve as grist for reflection and for learning experiments. Our idea is to construct episodic memories out of summaries of these logs, which could then be consulted for self-predictive purposes. Some kinds of knowledge that it should glean from these logs include expectations about the difficulty of the current task and how long it might take, whether or not it is thrashing (obviously not computable with 100 percent accuracy), and a context for referring to objects and tasks across sessions. Identifying the precursors to agent failure that can be used to trigger hot-swapping (such as memory becoming full) is also important.

First-Cut Companion Architecture

The Companion architecture is very much a work in progress. The first running version, from summer of 2004, is illustrated in figure 1. The Facilitator orchestrates the operation of the other agents. Interactions with a Companion are divided into sessions, with the reasoning work being carried out in the Session Reasoner. The Analogical Tickler, which uses MAC/FAC, tracks the current situation and maintains a short list of remindings that is available at any time. Each of these agents is running on its own node in the cluster. A number of agents also run on the user's machine. The Session Manager controls the startup and shutdown of agents on the user's machine; it also contacts the Facilitator to get the appropriate agents on the cluster started when a new session begins. A nuSketch system for sketching is wrapped as an agent as well, with the choice of nuSketch system being determined by the domain of the current session. As noted previously, the relational concept map system enables the entry of information that is best communicated by means other than sketching, for example, causal laws.

This configuration did not include several features that we think are crucial for a full Companion, some of which have subsequently been added. (We discuss the current state of the system later.) We have focused on this initial set of capabilities because they are the minimum that enable us to experiment with domain learning by accumulating examples. We have now conducted a number of experiments using this configuration, which has given us a baseline analogical reasoning model that we can use to judge the effectiveness of additional components. We mention some of these experiments next.

Experimental Domains

To ensure generality, we are working from the beginning in several domains. We have used two criteria in selecting these domains: (1) *Sources of complexity*. They must be broad, in contrast with the narrow domains in which systems operate comfortably now. (2) *Clear progress metrics*. Reasonably objective methods of measuring performance are important. When an externally validated performance standard is available, that is even better. We discuss each domain in turn.

Domain 1: Everyday Physical Reasoning

No existing AI system handles the breadth and flexibility of the kinds of reasoning that people do about the everyday physical world around them. While progress in qualitative reasoning has provided a powerful collection of representations to work with (Kuipers 1994), the reasoning techniques developed by that community, including ourselves, do not appear to be psychologically plausible (Forbus and Gentner 1997). The motto of "structure-mapping all the way down" for Companions encourages an approach that we think will be more plausible: within-domain analogical reasoning.

To provide a clear progress metric, we are starting with the kinds of problems found on the Bennett Mechanical Comprehension test. Figure 2 is an example of the kind of problem that one sees on this exam: Which way will it be easier to carry the rock in the wheelbarrow? (If equal, mark *C*). The Bennett Mechanical Comprehension test has been administered for more than 50 years, and is used to evaluate candidates for jobs involving mechanical skills. It is also commonly used by cognitive psychologists as an independent means to measure spatial abilities in experiments. It is an extremely broad test, covering topics such as statics, dynamics, heat flow, and electricity. However, the reasoning is mostly qualitative in nature, using comparative analysis (Weld 1990). Each of the two forms of the Bennett Mechanical Comprehension test has 68 questions.

In working with these problems, we used our *sketching Knowledge Entry Associate* (sKEA) (Forbus and Usher 2002) as part of the Companion's interface. We are sketching the pairs of situations in two dimensions, which is sufficient for all but a handful of the problems. A formal language is used to express the text of the question. The material to be learned includes (1) visual/conceptual mappings and conventions for depicting everyday objects, (2) modeling assumptions (for example, how parts of everyday objects map into qualitative mechanics formalisms), and (3) causal models (for example, that tall things with narrow bases tend to be less stable).

We have run two series of experiments so far. The first (Forbus, Usher, and Tomai 2005) has focused on learning visual/conceptual mappings (for example, the wheel/axle relationship in a wheelbarrow being a rotational connection). Given a corpus of sketches created by several users, can the system suggest conceptual relationships that might hold between elements of a sketch by analogy with prior sketches? The answer is yes, and with quite reasonable results—the system provides suggestions for roughly half of the visual/conceptual relationship questions that arose, with its accuracy ranging from 87 percent when the corpus was highly focused on a set of tightly related phenomena, to 57 percent, when a smaller corpus with more wide-ranging phenomena was used.[1]

The second set of experiments (Klenk et al. 2005) explored whether accumulating examples of physical principles could enable a system to solve Bennett Mechanical Comprehension test problems. Sketches illustrating physical principles were generated by students

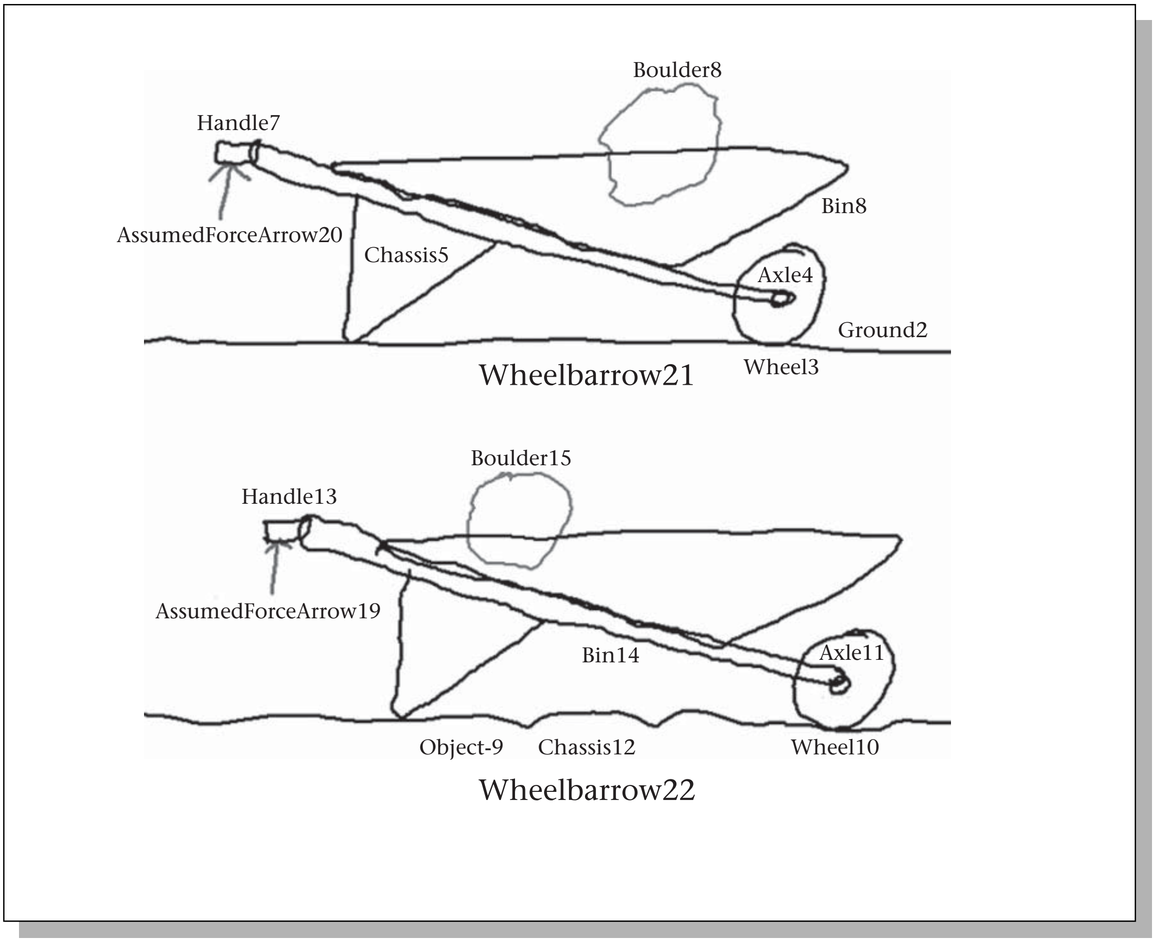

Figure 2. An Example of Everyday Physical Reasoning.

Which wheelbarrow would be easier to lift?

working from sentences describing a scenario (for example, "a crane is lifting a load"). These sketches include modeling assumptions and causal models. This additional information is entered in several ways. The visual/conceptual mapping interface provides information about relationships between items in the sketch; for example, the crane's wheels are directly above the ground. Annotations on the sketch are used to specify visual quantities, such as the distance from the load to the cab, as illustrated by the red line in figure 3. Finally, the relational concept map is used to express other kinds of statements, such as causal models; for example, the stability of the crane decreases as the distance from the cab to the load increases.

As already noted, the Bennett test is extremely difficult. We have focused on problems involving forces, selecting a subset of 13 problems to work with initially. To make things more difficult, we have examples drawn by a number of different people, and someone else draws the sketches representing the problems. For instance, figure 4 illustrates one problem. To solve this problem, the system retrieves analogs for the two systems being compared and uses them to create causal models for them. The newly elaborated problem situations are then compared with each other by analogy, using the correspondences of this analogy to provide the framework for doing a differential qualitative analysis.[2] Figure 5 illustrates the two situations with annotations constructed by means of analogical reasoning from the examples. The causal model determines that, to find stability, one needs to look at the distance from the cab to the load. Since this distance is defined visually, sKEA calculates which of these distances is larger (through the analogically inferred annotations). This information plus the causal model enables

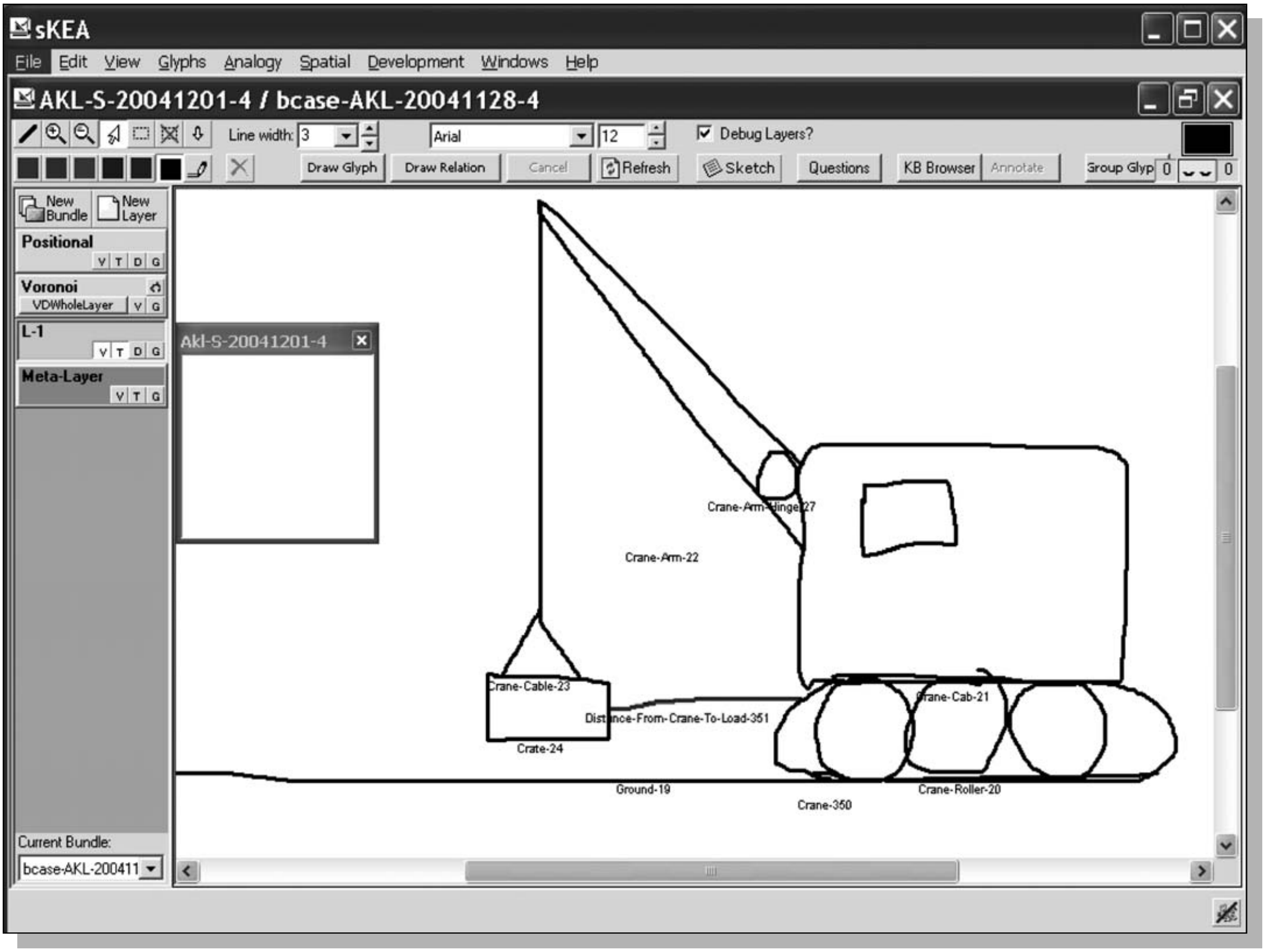

Figure 3. Examples of Physical Principles Are Described by Means of Sketches.

the system to conclude that the crane on the left will be less stable. So far, the best set of sketches yields correct answers of 10 out of 13 correct (77 percent), while the worst set of sketches yields only 2 correct answers (15 percent). Examining the failures is instructive. Most failures are due to people sometimes drawing systems very differently, such as leaving out parts or describing parts at different levels of detail within the knowledge base. This leads to some retrieval problems, but most of the problems arise in mapping. The ability to do rerepresentation (Yan, Forbus, and Gentner 2003), combined with a learned theory of depiction (such as when it is okay to leave parts out or include them), could lead to significant improvement.

We find both sets of experiments very encouraging. First, performance is already respectable, even with the simple baseline analogical learning model. Second, performance is not at ceiling; indeed, there is ample room for improvement. This means we can measure how well extensions to this model improve performance, such as using SEQL for generalization and using rerepresentation.

In addition to rerunning these experiments with extended analogical reasoning capabilities, we also are going to extend the library of examples and phenomena covered to attempt to solve the entire Bennett test. Once the system has learned enough to do well on the exam form we have access to, we will test it with a second form of the exam that has been sequestered, as one gauge of breadth. Another

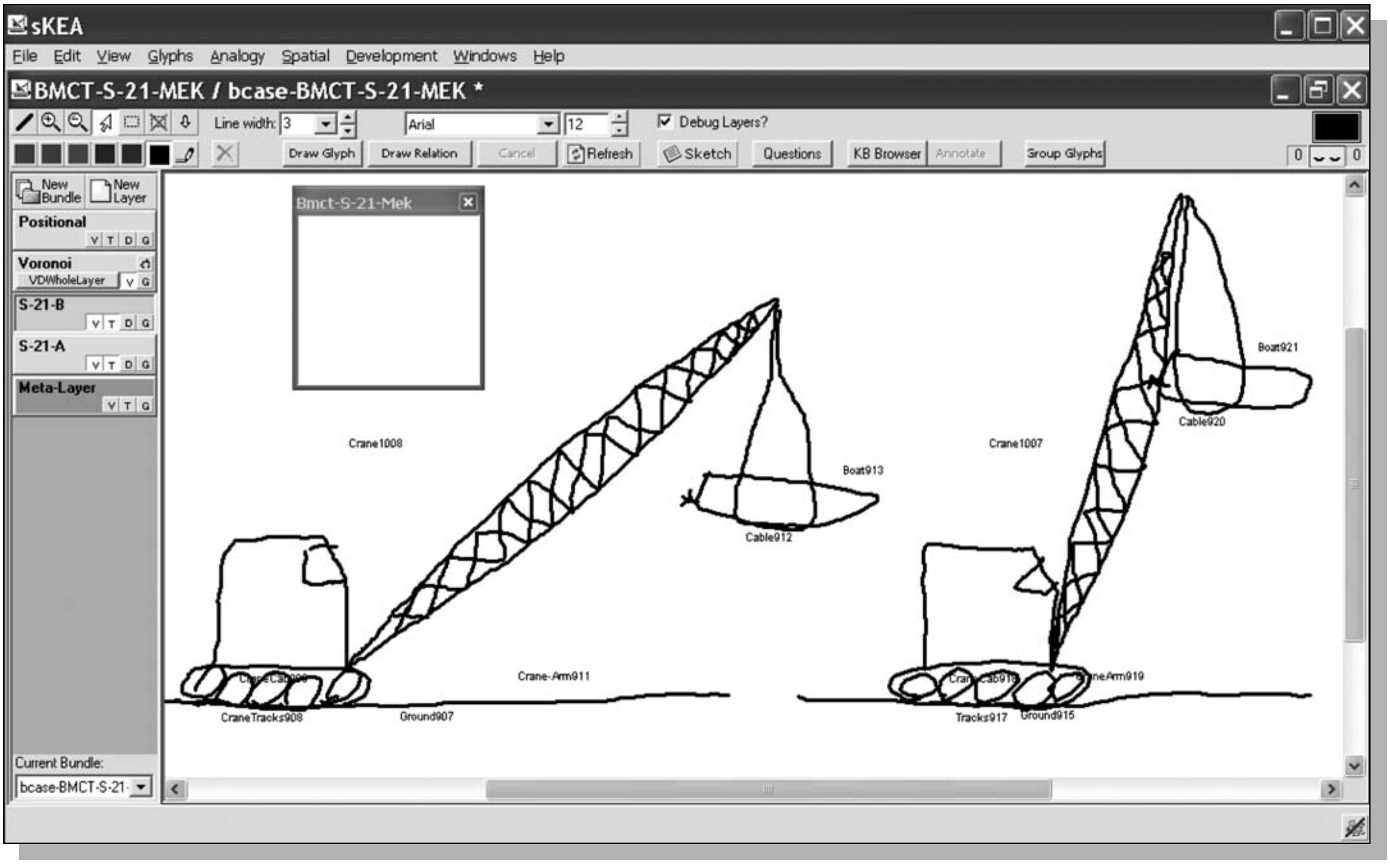

Figure 4. An Example Problem.

Question: Which crane is more stable?

kind of experiment planned is determining minimal sets of cases that suffice to give a particular level of performance on the exam, with and without generalization.

Domain 2: Tactical Decision Games

Tactical decision games (TDGs) are scenarios used by military personnel to hone their command skills (Schmitt 1994). A scenario is provided in a page or two of text describing the situation, accompanied by a sketch map illustrating the terrain and what is known about force layouts (figure 6). These scenarios are quite complex. Typically several answers are equally reasonable, given different assumptions about what the enemy is doing. In keeping with the aide-de-camp nature of Companions, we are not attempting to solve TDGs, but instead to provide several kinds of advice about the scenarios they embody. These include estimating enemy intent (that is, what are the bad guys up to?), identifying risks (that is, how might they prevent you from accomplishing your mission?), and identifying opportunities (that is, how might you go about carrying out your mission?).

We have obtained a corpus of TDGs courtesy of the *USMC Gazette* containing both a set of problems and several solutions to each problem. We encoded them using *nuSketch Battlespace* (Forbus, Usher, and Chapman 2003), our other sketch understanding system, as a component in the Companion interface. We performed a set of learning experiments to measure how effectively a Companion can incrementally acquire and apply new cases in such a complex and open domain. Unlike the everyday physical reasoning domain, evaluating learning and performance in tactical decision games is much less clear cut. We ultimately chose to compare performance against expert solutions, and to track improvement in that performance as a function of retrieval rank, in order to simulate the incremental accumulation of cases.

The first experiments exercised the Analogical Tickler to measure how accurately it could return precedents from a library of 16 cases,

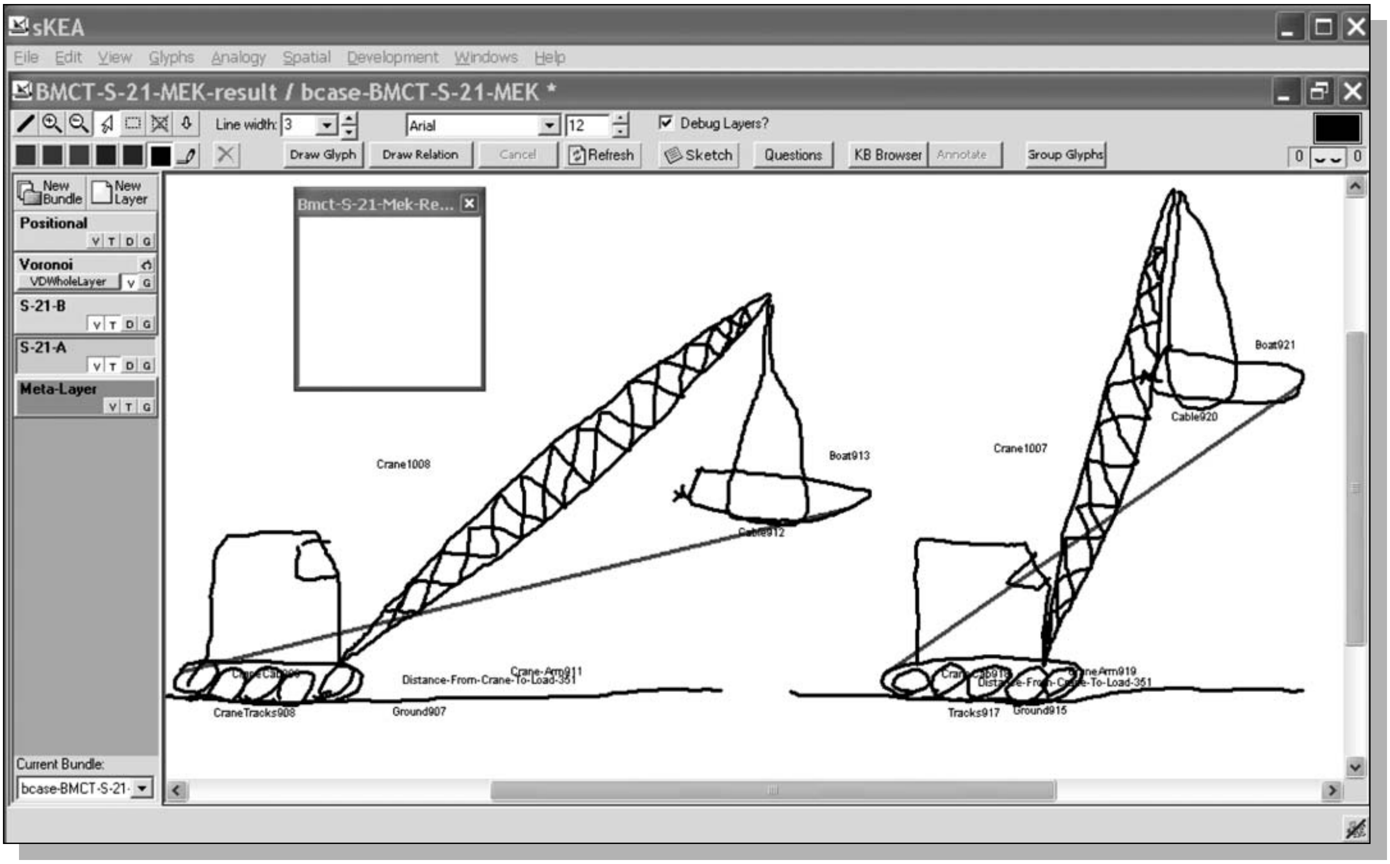

Figure 5. An Example Solution.

Causal models built by analogy are compared to derive the answer to the question posed in figure 4.

with respect to structural similarity and problem-solving performance (figure 7). These cases are reasonably large: 755 propositions on average, with the smallest being 345 and the largest being 1,233. We found that the tickler performed well, returning precedents within a few seconds and with an average error rate of 12 percent with respect to the best possible structural match. The next experiments tested reasoning and learning by proposing individual task assignments, an important component of solving a tactical decision game. We evaluated the quality of these suggestions by comparing them to tasks in the expert's solutions to the games, assigning credit based on similarity of the task type (for example, ambush, block, and so on), the target or object acted on, the unit assigned to perform the task, and the location or path followed in the task. Average performance on 12 problems improved 30 percent between solving from a structurally poor precedent to solving from the best of 16 precedents. As cases were added, the learning trend showed gradual improvement, though not perfectly monotonic. We find this encouraging because this simple learning technique is likely to be useful for self- and user-modeling, which are also domains for which there is no single correct answer.

Domain 3: FreeCiv

To carry on further with tactical decision games would require hiring military experts to evaluate results, and making the system friendly enough for them to work with for entering new detailed cases. While there is some evidence that this can be done (Pool et al. 2003; Rasch, Kott, and Forbus 2002), it would be a distraction. Thus the difficulty of evaluating results in tactical decision games has prompted us to add a third domain. We are currently using the strategy game FreeCiv as a testbed for Companion experiments. FreeCiv is an open-source version of *Civilization 2*. FreeCiv contains problems that are analogous to military tactical and strategic problems, but also involves economic decisions, diplomacy, and many other aspects that make it

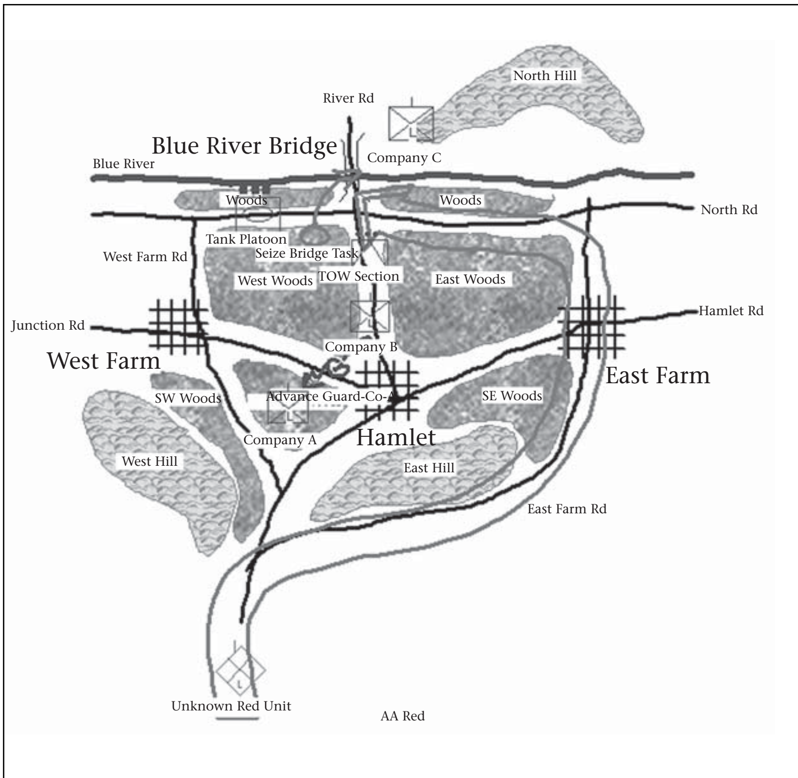

Figure 6. Sketch Map for a Tactical Decision Game.

a rich and complex arena. This will enable us to test Companions on a broader range of tasks and explore interactive learning over longer periods of time. An interactive game greatly simplifies case acquisition, affords more opportunities for user modeling, and may even enable us to experiment with cross-domain analogies. Our starting point has been a client-side AI player developed earlier at Northwestern (Houk 2004), which handles within-continent exploration and city placement decisions.

Related Work

Other cognitive architectures have tended to focus on skill learning (for example, Anderson and Lebiere [1998]; Laird, Newell, and Rosenbloom [1987]). By contrast, our focus is on conceptual understanding and learning. While the Companions architecture is strongly motivated by cognitive science results, particularly in its use of analogical processing, other aspects of it (such as the use of a distributed agent architecture) are motivated by the desire to carry out large-scale experiments on existing hardware, rather than representing hypotheses about human cognition.

There are a number of competing simulations of analogical matching, some of which also incorporate retrieval and representation construction. The connectionist models (for example, Eliasmith and Thagard [2001], Hummel and Holyoak [1997], and Larkey and Love [2003]) focus on neural plausibility, but cannot handle representations of the size used in this project, which casts doubt on their ultimate utility as psychological models. Other simulations that focus on representation construction (French [1995]; Mitchell [1993]) use domain-specific matching algorithms, unlike our use of domain-independent cognitive simulations. Simulations of analogy in problem solving also tend to use special-purpose matchers and retrieval mechanisms (for example, VanLehn and Jones [1993]). None of these simulations has been used as components in larger-scale systems, as SME and MAC/FAC have been.

Some aspects of our work have been inspired by Winston's pioneering work on using precedents in reasoning (Winston 1981) and research on case-based reasoning (compare Kolodner [1994]; Leake [1996]). CBR systems tend to use domain-specific and task-specific systems for matching and retrieval. One exception is PRODIGY-ANALOGY (Veloso and Carbonell 1993), which was the first broad-scale use of analogy in a cognitive architecture. Methods for identifying representations to improve matching, such as those proposed by M. Finlayson and P. Winston (Finlayson and Winston 2004), could possibly be used to improve Companion performance.

Future Work

We have made significant additions to the Companions system over the last year. A script-based Interaction manager, which centralizes interactions with the user for modeling purposes, has been installed. A first-cut version of the Executive, which monitors progress in the Session Reasoner and decides what should be done in response to user interactions and expectations, has also been installed. Similarly, we now have a SEQL agent that uses an extended version of SEQL that incorporates probabilities in its generalizations.[3] An HTN planner and execution monitor has been added to several agents, to handle strategies and tactics in FreeCiv, support plan recognition, and run the Executive. We are also decomposing our sketching software, so that the user interaction remains on desktops or tablets but the spatial reasoning will happen

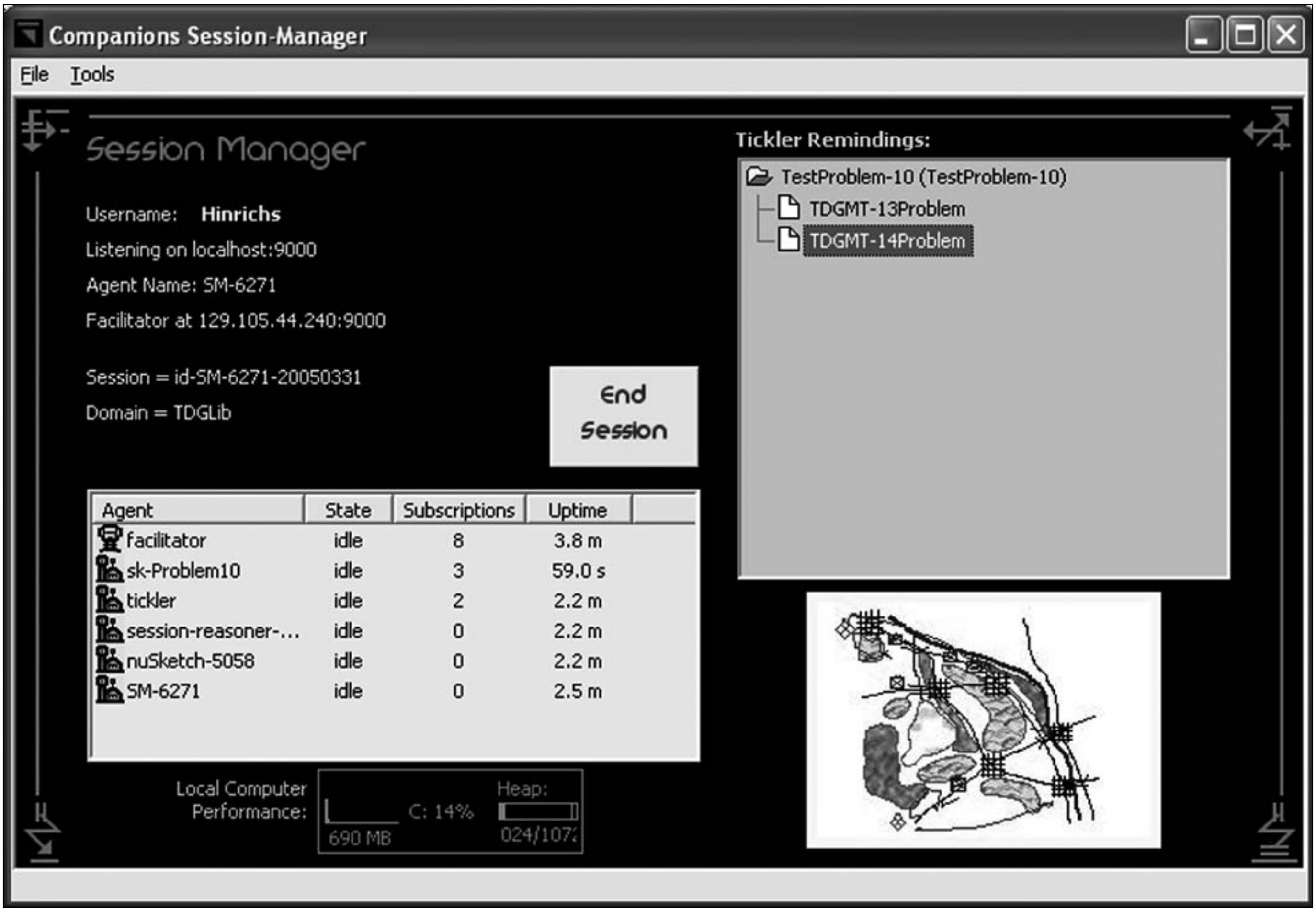

Figure 7. Analogical Tickler.

on a cluster. Our work as of this writing is focused on making these new capabilities work harmoniously together, for even better domain learning, learning user- and self-models, and increased autonomy.

The Companions project is still in its early stages, and most of the work lies ahead. Even so, it is already raising a variety of interesting questions, including the following:

Self-awareness. What contents, level of detail, and organization of logs are needed to support hot-swapping of components and learning better self-models?

Encoding. Some initial encoding of situations is needed to "prime the pump" for analogical retrieval. Learning good encoding strategies seems to be a key learning problem: Psychological results suggest that differences in encoding can account for many novice/expert differences (Chi, Feltovich, and Glaser 1981), for example.

Nonlinguistic multimodal communication. How far can we go with the combination of sketching and concept maps? How do these modalities impact dialogue models?

We hope that by answering these questions, we can help bring the goal of achieving human-level AI closer.

Acknowledgements

This research is supported by DARPA IPTO. The core Companions team (Greg Dunham, Hyeonkyeong Kim, Matthew Klenk, Emmett Tomai, Jeffrey Usher), with additional support from Patricia Dyck, Brian Kychelhahn, Kate Lockwood, made this research possible through their skill and tireless enthusiasm.

Notes

1. All numerical results mentioned in this article are statistically significant.

2. This use of analogy to frame comparative analysis questions (Weld 1990) is a novel contribution to qualitative reasoning.

3. This is research in progress conducted by Daniel Halstead.

References

Allen, J.; Ferguson, G.; and Stent, A. 2001. An Architecture for More Realistic Conversational Systems. In *Proceedings of the 2001 International Conference on Intelligent User Interfaces*. New York: Association for Computing Machinery.

Anderson, J.; and Lebiere, C. 1998. *The Atomic Components of Thought*. Mahwah, NJ: Erlbaum.

Cañas A. J.; Ford, K. M.; Brennan, J.; Reichherzer, T.; Hayes, P. 1995. Knowledge Construction and Sharing in Quorum. *World Conference on Artificial Intelligence in Education* 218–225. Brighton, UK: International AIED Society.

Chi, M. T. H.; Feltovich, P. J.; and Glaser, R. 1981. Categorization and Representation of Physics Problems by Experts and Novices. *Cognitive Science* 5(1): 121–152.

Cox, M. T. 2005. Metacognition in Computation. *Artificial Intelligence* 169(2): 104–141.

Eliasmith, C.; and Thagard, P. 2001. Integrating Structure and Meaning: A Distributed Model of Connectionist Mapping. *Cognitive Science* 25(2): 245–286.

Erol, K.; Nau, D.; and Hendler, J. 1994. UMCP: A Sound and Complete Planning Procedure for Hierarchical Task-Network Planning. In *Proceedings of the First International Conference on Artificial Intelligence Planning Systems*. San Francisco: Morgan Kaufmann Publishers.

Falkenhainer, B.; and Forbus, K. 1991. Compositional Modeling: Finding the Right Model for the Job. *Artificial Intelligence* 51(1–3): 95–143.

Falkenhainer, B.; Forbus, K.; and Gentner, D. 1989. The Structure-Mapping Engine: Algorithm and Examples. *Artificial Intelligence* 41(1): 1–63.

Finlayson, M.; and Winston, P. 2004. A Model of Analogical Retrieval Using Intermediate Futures. In *Proceedings of the 26th Annual Meeting of the Cognitive Science Society*. Wheat Ridge, CO: Cognitive Science Society.

Forbus, K. 1984. Qualitative Process Theory *Artificial Intelligence* 24(1–3): 85-168.

Forbus, K.; Carney, K.; Sherin, B.; and Ureel, L. 2004. VModel: A Visual Qualitative Modeling Environment for Middle-SchoolStudents. In *Proceedings of the Sixteenth Conference on Innovative Applications of Artificial Intelligence*. Menlo Park, CA: AAAI Press.

Forbus, K.; Ferguson, R.; and Gentner, D. 1994. Incremental Structure-Mapping. In *Proceedings of the 26th Annual Meeting of the Cognitive Science Society.* Wheat Ridge, CO: Cognitive Science Society.

Forbus, K.; and Gentner, D. 1997. Qualitative Mental Models: Simulations or Memories? Paper presented at the Eleventh International Workshop on Qualitative Reasoning, Cortona, Siena, Italy, June 1997.

Forbus, K.; Gentner, D.; and Law, K. 1995. MAC/FAC: A Model of Similarity-Based Retrieval. *Cognitive Science* 19(2)(April-June): 141–205

Forbus, K. D.; and Kuehne, S. E. 1998. RoboTA: An Agent Colony Architecture for Supporting Education. In *Proceedings of the International Conference on Autonomous Agents,* 455–456. New York: Association for Computing Machinery.

Forbus, K.; Mostek, T.; and Ferguson, R. 2002. An Analogy Ontology for Integrating Analogical Processing and First-Principles Reasoning. In *Proceedings of the Thirteenth Conference on Innovative Applications of Artificial Intelligence*. Menlo Park, CA: AAAI Press.

Forbus, K.; and Usher, J. 2002. Sketching for Knowledge Capture: A Progress Report. In *Proceedings of the Intelligent User Interfaces Conference*. New York: Association for Computing Machinery.

Forbus, K.; Usher, J.; and Chapman, V. 2003. Qualitative Spatial Reasoning about Sketch Maps. In *Proceedings of the Fourteenth Conference on Innovative Applications of Artificial Intelligence*. Menlo Park, CA: AAAI Press.

Forbus, K.; Usher, J.; and Tomai, E. 2005. Analogical Learning of Visual/Conceptual Relationships in Sketches. In *Proceedings of the Twentieth National Conference on Artificial Intelligence*. Menlo Park, CA: AAAI Press.

French, R. 1995. *The Subtlety of Similarity*. Cambridge, MA: The MIT Press.

Gentner, D. 1983. Structure-Mapping: A Theoretical Framework for Analogy. *Cognitive Science*, 7(2): 155–170.

Gentner, D.; and Markman, A. 1997. Structure Mapping in Analogy and Similarity. *American Psychologist* 52(1): 45–56

Horvitz, E.; Kadie, C. M.; Paek, T.; and Hovel, D. 2003. Models of Attention in Computing and Communication: From Principles to Applications. C*ommunications of the Association for Computing Machinery (CACM)* 46(3): 52–59.

Houk, P. 2004. A Strategic Game Playing Agent for FreeCiv. Technical Report NWU-CS-04-29, Northwestern University, Computer Science Department. Evanston, IL.

Hummel, J. E.; and Holyoak, K. J. 1997. Distributed Representations of Structure: A Theory of Analogical Access and Mapping. *Psychological Review* 104(3): 427–466.

Kim, H. 1993. Qualitative Reasoning about Fluids and mechanics. Ph.D. dissertation and ILS Technical Report, Northwestern University, Computer Science Department. Evanston, IL.

Klenk, K.; Forbus, K.; Tomai, E.; Kim, H.; and Kyckelhahn, B. 2005. Solving Everyday Physical Reasoning Problems by Analogy Using Sketches. In *Proceedings of the Twentieth National Conference on Artificial Intelligence*. Menlo Park, CA: AAAI Press.

Kolodner, J. L. 1994. *Case-Based Reasoning*. San Mateo, CA: Morgan Kaufmann Publishers.

Kuehne, S.; Forbus, K.; and Gentner, D. 2000. SEQL: Category Learning as Incremental Abstraction Using Structure Mapping. In *Proceedings of the 22nd Annual Meeting of the Cognitive Science Society.* Wheat Ridge, CO: Cognitive Science Society.

Kuehne, S.; Gentner, D.; and Forbus, K. 2000. Modeling Infant Learning Via Symbolic Structural Alignment. In *Proceedings of the 22nd Annual Meeting of the Cognitive Science Society.* Wheat Ridge, CO: Cognitive Science Society.

Kuipers, B. 1994. *Qualitative Reasoning: Modeling and*

Simulation with Incomplete Knowledge. Cambridge, MA: The MIT Press.

Labrou, Y.; and Finin, T. 1997. A Proposal for a New KQML Specification. Technical Report TR CS-97-03, University of Maryland Baltimore County, Baltimore, MD.

Laird, J.; Newell, A.; and Rosenbloom, P. 1987. SOAR: An Architecture for General Intelligence. *Artificial Intelligence* 33(1): 1–64.

Larkey, L.; and Love, B. 2003. CAB: Connectionist Analogy Builder. *Cognitive Science* 27(5): 781–794.

Leake, D. 1996. *Case-Based Reasoning: Experiences, Lessons, and Future Directions*. Menlo Park, CA: AAAI Press.

McDermott, D. 2001. *Mind and Mechanism*. Cambridge, MA: The MIT Press.

Mitchell, M. 1993. *Analogy-Making as Perception: A Computer Model*. Cambridge, MA: The MIT Press.

Mostek, T.; Forbus, K.; and Meverden, C. 2000. Dynamic Case Creation and Expansion for Analogical Reasoning. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*. Menlo Park, CA: AAAI Press.

Nielsen, P. E. 1988. A Qualitative Approach to Rigid Body Mechanics. Tech. Rep. No. UIUCDCS-R-88-1469; UILU-ENG-88-1775. University of Illinois at Urbana-Champaign, Department of Computer Science, Urbana, IL.

Novak, J. D.; and Gowin, D. B. 1984. *Learning How to Learn*. New York: Cambridge University Press.

Pool, M.; Russ, T.; Schneider, D.; Murray, K.; Fitzgerald, J.; Mehrota, M.; Schrag, R.; Blythe, J.; Kim, J.; Chalupsky, H.; and Miraglia, P. 2003. Evaluating Expert-Authored Rules for Military Reasoning. In *Proceedings of the Second International Conference on Knowledge Capture*. New York: Association for Computing Machinery.

Rasch, R.; Kott, A.; and Forbus, K. 2002. AI on the Battlefield: An Experimental Exploration. In *Proceedings of the Fourteenth Innovative Applications of Artificial Intelligence Conference*. Menlo Park, CA: AAAI Press.

Schmitt, J. F. 1994. *Mastering Tactics: A Tactical Decision Game Workbook*. Quantico, VA: Marine Corps Association.

Thomere, J.; Rodriguez, A.; Chaudhri, V.; Mishra, S.; Eriksen, M.; Clark, P.; Barker, K.; and Porter, B. 2002. A Web-Based Ontology Browsing and Editing System. In *Proceedings of the Fourteenth Innovative Applications of Artificial Intelligence Conference*. Menlo Park, CA: AAAI Press.

VanLehn, K.; and Jones, R. 1993. Better Learners Use Analogical Problem Solving Sparingly. In *Proceedings of the Tenth International Conference on Machine Learning*, 338–345. San Francisco: Morgan Kaufmann Publishers.

Veloso, M.; and Carbonell, J. 1993. Derivational Analogy in PRODIGY: Automating Case Acquisition, Storage, and Utilization. *Machine Learning* 10: 249–278.

Weld, D. 1990. *Theories of Comparative Analysis*. Cambridge, MA: The MIT Press.

Winston, P. 1981. Learning New Principles from Precedents and Exercises: The Details. MIT AI Laboratory Memo #632, Massachusetts Institute of Technology, Cambridge, MA.

Yan, J.; Forbus, K.; and Gentner, D. 2003. A Theory of Rerepresentation in Analogical Matching. In *Proceedings of the 25th Annual Conference of the Cognitive Science Society.* Wheat Ridge, CO: Cognitive Science Society.

Kenneth D. Forbus is the Walter P. Murphy Professor of Computer Science and Professor of Education at Northwestern University. His research interests include qualitative reasoning, analogy and similarity, sketching and spatial reasoning, cognitive simulation, reasoning system design, articulate educational software, and the use of AI in computer gaming. He received his degrees from the Massachusetts Institute of Technology (Ph.D. in 1984). He is a Fellow of the American Association for Artificial Intelligence and the Cognitive Science Society, and serves on the editorial boards of *Artificial Intelligence, Cognitive Science,* AAAI Press, and the Advisory Board of the *Journal of Game Development*. His e-mail address is forbus@northwestern.edu.

Thomas R. Hinrichs is a research associate professor at Northwestern University. He received his B.S. from Cornell University and his Ph.D. from Georgia Tech in 1991. His research interests include analogy, machine learning, and design. His e-mail address is hinrichs@cs.northwestern.edu.