Report on the NSF-funded
Workshop on Taskability
(Revised Title: Interactive Task Learning)

May 12-13, 2013
University of Michigan
John E. Laird
Report submitted July 16, 2014

Contributors:

Attendees:

John Anderson (Carnegie Mellon University), Ken Forbus (Northwestern University), Kevin Gluck (Air Force Research Laboratory), Chad Jenkins (Brown University), John Laird (University of Michigan), Christian Lebiere (Carnegie Mellon University), Dario Salvucci (Drexel University), Matthias Scheutz (Tufts University), Andrea Thomaz (Georgia Institute of Technology), Greg Trafton (Naval Research Laboratory), Bob Wray (Soar Technology)

Scribes:

Shiwali Mohan (University of Michigan), James Kirk (University of Michigan)

NSF Award Id: 1419590
PI: John E. Laird, University of Michigan

This is the final report for the "Workshop on Taskability" sponsored by NSF. The workshop was held in Ann Arbor, MI on May 12-13, 2014. The workshop brought together senior researchers interested in interactive task learning – computational systems that can learn completely new tasks from interaction with human teachers. The goal of the workshop was to explore this topic as a potential research problem and to take the first steps to build a community of researchers that would take the lead in establishing it as a research area. The body of this report introduces the rationale for the workshop and then provides descriptions of the discussions that were held at the workshop. The overall conclusion of the workshop is that this is an area of research that should be aggressively pursued. All of the participants were deeply engaged in the discussions and committed to continuing development of this area of research. We generated both concrete and conceptual steps to pursue in the immediate future. One concrete result of the workshop was that we agreed to call this area of research "Interactive Task Learning" (ITL) and that label is used throughout the rest of the report (in place of "Taskability").

There was enthusiasm for follow-on workshops to:
1. Identify important research precursors that are needed to support an expansion of research in interactive task learning.
2. Identify the most important research problems that need to be addressed in interactive task learning.
3. Develop more specific plans in specific domains, such as assistive robotics, games, education, cognitive science research, and training.
4. Raise awareness of interactive task learning as an important and exciting research area.
5. Further refine the definitions, desiderata, and analysis of task learning.
6. Directly address issues that arise when funding agencies considering starting new research programs. For example, it would be useful to hold a workshop (sort of a mini DARPA ISAT meeting) to discuss and to develop answers to the Heilmeier Catechism – a set of questions that DARPA uses when considering starting a new research program.

Outline of remainder of this report:

1. Introduction
2. Interactive Task Learning: Definitions and desiderata
3. Domains and associated applications for ITL
4. Characterizations of ITL dimensions and their variations
5. Community building for future research in ITL

Appendix I: Final schedule that was followed at the workshop
Appendix II: Annotated and extended bibliography
Appendix III: Additional references

Section 1. Introduction

One of the uniquely human characteristics is that we are not limited to a fixed set of innate/preprogrammed tasks – we quickly learn novel tasks through language and other forms of natural communication, and once we learn them, we learn to perform them better. We learn to play some new games in just a few minutes; we learn how to use new devices such as smart phones, computers, and industrial machinery; and we can learn how to help a disabled family member with their everyday tasks, adapting to their needs over time. Advances in Artificial Intelligence, Cognitive Science, and Robotics are leading us to a future populated with autonomous systems that will have the cognitive and physical capabilities to perform a wide variety of tasks, with applications across science, health care, business, the military, home, and entertainment. But how will these agents learn the complex tasks we want them to perform? The goal of this workshop was to take the first steps in establishing a field that studies and develops the science and technology to support *interactive task learning* - independent intelligent artificial agents that learn new tasks through natural interactions with humans. This is an extremely ambitious problem to tackle, but recent progress in many of the related fields suggest that now is the time to make a cooperative and coordinated attack on interactive task learning.

Understanding how an agent acquires new tasks through natural interaction is a fundamental unsolved problem in AI and Cognitive Science. Pursuing it will increase our understanding of how an agent converts an externally specified task description into efficient executable procedural knowledge that is incrementally and dynamically integrated with existing knowledge. It will also increase our understanding of how essentially all the capabilities we associate with cognition work together, including extracting task-relevant meaning from perception, task-relevant action, language processing, dialog and interaction management, integrated knowledge-rich relational reasoning, problem solving, learning, and metacognition. This integration is in contrast to most current research in AI and Cognitive Science, which has become more and more fragmented and focused on narrow problems.

In addition, if we can extend our understanding of task learning to include how humans learn new tasks, it will be a major advance for cognitive modeling, and potentially training and education. Understanding task learning will not only make modeling become much easier and faster, it could provide insight into how to structure task teaching to make task learning easier and faster for people.

Interactive task learning agents will fundamentally change the way we interact with intelligent agents and robotic systems across science, health care, industry, rescue/military, home, and entertainment. No longer will these systems be limited to preprogrammed tasks. Human users will be able to teach them new tasks dynamically and to help improve their performance through instruction and demonstration. This has obvious applications in health care, where co-robots will need to be customized to the needs of their patients; in industry, where the tasks performed by industrial robots will change dynamically and be unique to a specific work place; for rescue and military applications, where robots will need to be dynamically tasked to novel missions in new environments; and for the home, where service robots will need to be extended and customized to their users' specific needs. There are also many applications in developing intelligent support software for business and intelligence analysis, where creating and maintaining models of dynamically unfolding events based on incomplete, inaccurate, and deceptive information requires updating information gathering and integration strategies; in entertainment, where developing rich virtual characters that can be taught by players will open new dimensions of immersion; and in education, where ever-changing curriculum and needs of learners will require software tutors and coaches that are extensible without reprogramming.

Section 2. Interactive Task Learning: Definitions and Desiderata

Throughout the history of AI, there have been many research efforts on learning different aspects of a task, such as how to perform a task better; however, very few of them have focused on learning the concepts that define the task, and different types of tasks are defined in different ways. For example, games and puzzles are usually defined by the environment in which the game take places, the objects that are used, the legal actions that can be taken, and the goal to be achieved[1]. Assembly tasks, such as cooking a dinner, or building a piece of furniture, are usually defined by a set of actions to be taken along with constraints on the final product. Classification tasks involve learning labels for sets of data. However, there has been little research done on how to learn the sets of actions or goal descriptions that define a task. Most learning research *assumes* these aspects of a task are already known to an agent, and an agent's learning task is to learn how to perform a task well. In contrast, the core of interactive task learning is to learn these concepts so that a task can be attempted. Learning to perform a task well is also in the purview of task learning, but that and other types of learning are only part of the story.

Our focus is on *interactive* task learning, where an AI agent learns a task from another entity, typically a human teacher. The interaction can include language, where the teacher describes the task, or the interaction could include the teacher demonstrating the task, and the agent learning by observing what the teacher is doing. The agent is thus learning during the interaction, possibly asking questions, or asking for new examples of the task. The agent might also learn autonomously from its own experiences with a task; however, it must have the ability to learn from natural interactions with another entity.

To further clarify the goals of research on interactive task learning, we developed a set of desiderata. These desiderata provide dimensions for evaluating and comparing ITL agents. These desiderata are for the "ultimate" ITL agent, and not all of them will apply to specific ITL agents that are developed for specific domains or for limited sets of tasks.

---

[1] See Section 4 for more details about the components of task definitions for different types of tasks.

D1. Learning competent

The primary goal of an interactive task learner is to learn a task from its interactions with a teacher and from its own experiences. It must have the necessary reasoning and learning capabilities to interpret instructions, map them onto the current situation, extract information about the task, generalize from examples and demonstrations, and store experiences in its memories for future use.

D1.1. Reasoning competence in task learning

The learning agent has reasoning capabilities so that it maximizes the value of the knowledge (in whatever form) provided by the teacher. It is a "smart" student that has language, reasoning, and problem solving abilities so that it can figure out "obvious" aspects of a task on its own. It takes advantage of its background knowledge that it has acquired from its own experience or other preexisting knowledge bases, so that it does not have to be taught everything. It has other capabilities that aid interaction, such as being able to explain why it is doing what it is doing, and it is capable of testing out a partial understanding of the instruction and debugging that understanding through interactive processes such as hypothesis testing or asking clarification questions.

D1.2. Learning competence in task learning

The learning agent has the necessary learning mechanisms to incorporate and potentially generalize and store the knowledge it is exposed to for use in the future. The agent can learn everything it needs beyond its initial programming in order to be competent in a task (D2: see below). A human teacher never needs to "escape" to direct programming to modify internal data structures (brain surgery). Moreover, the agent accumulates knowledge and new tasks over potentially long periods of time, such as years. This knowledge is not just a set of procedures for task execution, but is also more general conceptual knowledge about the structure of tasks and its components. This enables the agent to transfer knowledge learned from similar tasks to new tasks, eliminating the need for redundant interactions. It has learning capabilities (both active and passive) so that it learns from its own experiences, improving its performance on tasks and enriching its conceptual structures. It also can identify what it does not know, so that it can ask for additional information from its teacher.

One research issue in ITL is determining the primitive reasoning and learning capabilities and background knowledge that are necessary to support learning competence.

D2. Task competent

Task competence is directly related to the concept of "understanding" as defined by Simon (1977): *"S understands task T if S has the knowledge and procedures needed to perform T."* The purpose of interactive task learning is for an agent to learn the knowledge necessary to perform well on a new task. However, just as learning competence can be enhanced by background knowledge and other learning capabilities, task competence can be enhanced by additional background knowledge and many forms of task-independent reasoning, problem solving, and planning capabilities that an agent can marshal to apply the task knowledge it has learned. In the limit, task competence includes all the task management abilities required in a general autonomous intelligent agent, such as supporting the pursuit of multiple tasks, interrupting low priority tasks with higher priority tasks, and resuming suspended tasks. Furthermore, an agent should be able to acquire and learn additional knowledge so that it achieves mastery of the task. Achieving mastery often requires more than the simple accumulation of knowledge, but can also require learning new representations and strategies.

D3. Task general

An ITL agent learns a diverse set of tasks, which require a diverse set of concepts (objects, categories, relations), procedures (hierarchical, recursive, interruptible), and goals (achievement, maintenance, process). Although there may be ITL agents that are specialized to learning specific types of tasks (such as puzzles and games, or procedure-based tasks), the ultimate goal is to understand what is required to learn all types of tasks. Another goal of the research is to understand the limits of the interaction-based approach to task learning. There very well may be tasks that are best learned through other approaches, such as non-interaction-based experience.

D4. Easy to teach

Beyond competence and generality, ITL agents should be easy to teach, so that human teachers do not require extensive training, and so that interacting with the agent is not laborious.

D4.1 Accessible communication

Communication with an agent is unconstrained and natural for an untrained teacher. Accessibility can be increased by supporting multiple modalities, such as language, gestures, and diagrams, as well as through allowing reference to shared background knowledge and analogies. Communication should not require extensive knowledge of the internal mechanisms and representations of agent on the part of human teacher.

D4.2 Efficient communication

The representation of information required to communicate the task is concise and efficient. This includes minimizing the amount of information that has to be communicated for a new task, as well as minimizing the actual form of that information (such as the number of words). The amount of information should approach the efficiency of natural interactions between humans, which is significantly less than required in current approaches, such as programming languages or even specialized task-specification languages.

To support accessible and efficient communication, the teacher should be able to easily construct an abstract model of the agent’s processing, reasoning, and existing knowledge. A teacher with a good model of a student can skip instructions that the student can infer on its own, and focus on those aspects of the task that will be challenging for the student to learn. This implies that the agent can explain its reasoning and performance when necessary. One additional implication might be that the interaction should be modeled on human-to-human interaction, such as the communication that occurs between teachers and students, between trainers and trainees, and between teammates.

D4.3 Robust to communication errors

The agent should be robust to errors in the instructions so that the teacher does not have to provide perfect instructions. There might be many different ways of supporting robustness, including having the agent verify the correctness of instructions in some way (such as through internal simulation), or trying out the instruction and allowing the teacher to correct any errors. Being able to correct knowledge errors overlaps with an agent being learning competent (D1). Recent research on the nature of robustness (Gluck et al, 2012), a domain-general methodology for quantifying robustness and stability (Walsh, Einstein, & Gluck, 2013), and mechanisms that produce robust cognitive systems (Walsh & Gluck, in press) provides a conceptual and methodological foundation for the formal assessment of the degree of robustness achieved by systems capable of interactive task learning, as well as implementation guidance for producing greater robustness.

D5. Efficient execution

There should be no inherent computational penalty to learning a task interactively as opposed to through other approaches. Thus, at some point after an agent acquires a new task, the agent's execution of the task approaches the efficiency achieved when the task is programmed by hand in the same underlying architecture. Achieving efficient execution may require practice, as well as internal reasoning and analysis beyond the simple translation of task information into the agent's internal representations.

D6. Integrates well with agent activities

Our vision for the ultimate ITL agent is one where task learning is "just another task" that occurs in concert with all of the other agent's activities and tasks. Task learning is not a separate program that runs independently of an agent and that compiles knowledge off line and then feeds it to the agent. Nor is an agent in some special mode when it is learning a task so that it is oblivious to its other activities. Task learning happens while the agent is actively trying to perform a task, and although some aspects might require extensive internal processing or even external practice, those activities don't disrupt the agent's ongoing existence. If task learning is "just another task," it might be possible for an agent to improve its task learning abilities through instruction from a teacher.

D7. Domain/application-specific desiderata

For many domains and uses of interactive task learning, there will be additional task learning desiderata. For example, in cognitive modeling, the agent's behavior in performing the tasks that it learns should model human behavior, and the agent's learning of a task should model human learning behavior – what is easy for a human to learn should be easy for the agent, and what is

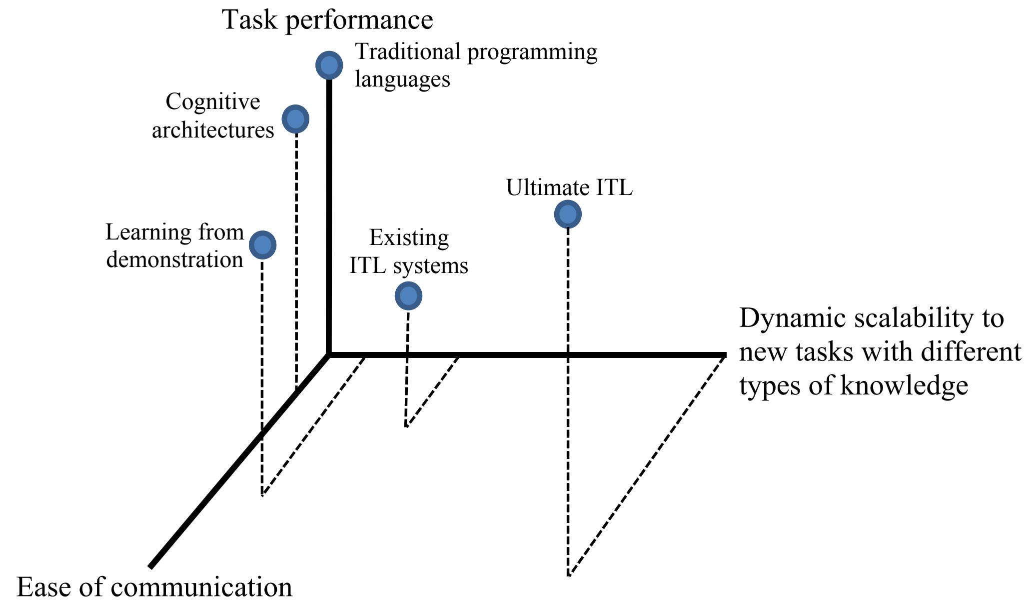

Figure 1: Comparison of Task Specification Techniques using Major ITL Desiderata

hard for a human should be hard. For other applications, it will be necessary that the agent verify that the new tasks it learns do not violate some pre-specified constraints, such as rules of engagement.

Desiderata summary

These desiderata allow us to compare our goal for ITL to the capabilities of other approaches. Figure 1 is an attempt to visualize this, where we have combined a subset of the desiderata into three dimensions. The vertical dimension is a measure of task competence (D2) and efficient execution (D5) – the quality of task performance after learning. The horizontal dimension measures the ability to dynamically scale to new tasks. This is the essence of competent task learning (D1) together with the generality to learn many different tasks with different types of knowledge (D3). The final dimension is the ease of communicating new knowledge (D4), which involves teaching for an ITL system, but programming for more traditional approaches.

Traditional programming languages support the development of systems with high task performance and efficient execution; however, they are difficult to develop and they do not inherently support dynamic extension to new tasks. Cognitive architectures provide some important capabilities (such as memory structures, decision making and learning mechanisms) that improve the ease of development and help support interactive task learning while maintaining high task performance; however, they do not inherently provide dynamic learning of new tasks without additional knowledge. Learning from demonstration systems usually have good task performance on the tasks they are taught and they eliminate the need for programming, greatly simplifying the communication of knowledge. However, they are restricted to learning only a few types of tasks, and there are only limited types of knowledge that are easily communicated through demonstration. There are some existing ITL systems (see the annotated bibliography) that can learn multiple tasks, although their efficiency and final performance is not yet to the level of hand-programmed systems. They also do not have the flexibility or ease of teaching that we desire for the ultimate ITL system, which is at the extreme for all three dimensions.

There are also systems that support human-interaction for commanding, extending, and correcting behavior through language or gestures, but they do not learn new tasks. For example, there are personal assistants (such as SIRI or Cortana) and systems for commanding mobile robots. Some of these allow the use of complex control constructs; while others allow a human to use language to extend or correct behavior. However, these do not learn completely new tasks but instead just improve or customize existing task performance.

One of the successes of the workshop was general agreement on the desiderata. One future goal is to further refine the desiderata so that they are used by researchers in the field to frame their claims and evaluations of interactive task learning systems.

Section 3. Domains and Associated Applications for ITL Research

In this section we describe possible domains and application areas. For a given domain, there might be multiple applications. For example interactive mobile robots can be used for many different applications in medical care, industry, the military, and the home. Thus, there appear to be many "vertical" slices of domains, each with clusters of tasks, where interactive task learning could have a positive impact. Each of these slices will not require comprehensive solutions to the complete diversity of dimensions that characterize interactive task learning as described in Section 4.

Games and Puzzles

Games and puzzles have many properties that make them attractive as testbeds for initial investigations of interactive task learning. Most have well defined actions, rules, and goals that are embedded in spatial domains. They also are tasks where a formulation of the task can be learned independent of a specific strategy or policy for task performance. For example, when learning chess, an agent can learn about the board, pieces, their movements, and the rules of the game without ever learning a strategy. For many other tasks, such as cooking, learning to perform the task is intertwined with what the task is (baking a chocolate cake) and how the task is achieved (the individual steps of measuring and combining ingredients). Games and puzzles also have the advantage that there are many pre-existing examples that have diverse types of written rules. In addition, people are familiar with teaching and learning games, making it natural for people to teach computers new games.

The first attempt at task learning was the UNDERSTAND project (Simon & Hayes, 1976), which attempted to build a system that could learn different isomorphs of the Tower of Hanoi puzzle. Simon and Hayes showed that differences in task specification led to differences in task formulation in humans, and that similar differences would arise in UNDERSTAND. Barbu, Narayanaswamy, and Siskind (2010) and Kaiser (2012) describe systems that learn a few simple games (such as Tic-Tac-Toe) through observation. Kirk and Laird (2013) and Hinrich and Forbus (2013) describe systems that learn multiple new games from restricted natural language instructions, and in the case of Hinrichs and Forbus also through demonstrations using sketches. All of these approaches differ from the General Game Playing competition, where programs get a complete, formal definition of a game, and then attempt to create high-performance game players through batch processing and without any human interaction.

Challenges:

Although there has been success with simple games with limited number of objects, it is unclear whether current techniques can scale up to more complex games. Some of the dimensions of complexity include the number of movable pieces and places (such as in chess or go), the number and complexity of rules and their interactions (such as in strategy games), the number and types of relevant spatial relations (such as terrain in Civilization), and the mental models required of other players. One weakness of most puzzles and games in comparison to other domains is that there is only limited dynamics in the environment. In most puzzles, the player is responsible for all environmental changes, while in most games, the changes come from the player and the opponents.

Possible Shared Challenge Problems:

1. A series of simple puzzles and games. Kirk and Laird (2013) have collected eleven such games, and this set could be greatly expanded to include many common games, including chess, go, backgammon, and so forth. These would provide common challenge problems for many groups to work on.
2. Freeciv. Forbus has been using Freeciv (an open source version of the turn-based strategy game Civilization) for research. It involves complex rules, terrain, and multiple players and is well suited as a challenge problem for ITL systems.
3. Video games. There are a variety of video games that are freely available and can easily be interfaced to from ITL systems. These include classic Atari games, such as Frogger and Space Invaders, but other games such as Infinite Mario. One "disadvantage" of these games is that they might not stress an ITL system because there is usually a single goal of getting the most points, and the available actions are determined by the environment. Thus, for many of these games, there is little need for interactive learning.

Collaborative Robots

Collaborative robots are an obvious domain for interactive task learning as rarely is it possible to predict all of the different tasks a human user will want a robot to do. Moreover, research in mobile, multipurpose robots has greatly expanded in recent years, so that there are more and more robots with the capabilities necessary for performing many different tasks interactively with a human. Application areas include domestic, medical, industrial, and military robots.

There is growing research in dynamic teaching of robots, and many of these systems are covered in Appendix II. Much work has focused on learning from demonstration (Argall et al. 2009; Nicolescu & Mataric 2003; Bentivegna, Atkeson, & Gordon 2004; Grollman & Jenkins 2009; Barbu, Narayanaswamy & Siskind 2010; Chao, Cakmak & Thomaz 2011). This work focused on either remote control of the robot or physically manipulating the robot directly to teach it specific tasks. Most of this work did not use language or used only scripted language models. Demonstration learning allowed quite natural instruction (e.g., a person moving herself or the robot) and the instruction environment was very similar to the testing environment, which not only simplified the problem space, but increased the overall success of the system.

Later work has built on this foundation and has greatly increased the natural language capabilities of the instruction, allowing inferencing, generalization, and non-scripted language (Dzifcak et al. 2009; Scheutz et al. 2011). Other work has allowed modification and extension to learned behaviors (Weitzenfeld, Ejnioui, Dominey 2010; Cantrell et al. 2012). Many other systems provide varying levels of support for interactive task learning (Huffman & Laird 1995; Lauria et al. 2001; Rybski et al. 2008, Mohan et al. 2012, Petit et al., 2013).

Active Learning is technique in which the language of questions and the semantics of the task need to be addressed in order for a robot to pose questions to a human supervisor. Some work assumes that a common language is known about the task and domain semantics in order for the robot to ask questions about specific actions and states (e.g., Chao et al. 2010), while other research makes use of the context of the interaction to ask questions physically without the need for specific language about actions or features of the domain (Chernova & Veloso 2009, Cakmak & Thomaz 2012).

Similar recent developments have increasingly incorporated crowdsourcing for collecting demonstration data (Crick et al. 2011, Toris et al. 2014) for faster and more adaptable robot learning. Such methods often involve web-enabled robots (Osentoski et al. 2012) that can be time-shared by multiple projects and access by users through common web browsers.

Challenges:

1. Integration and empirical systems: Robotics requires an integrated architecture and functioning systems that have at least partial solutions for object identification, manipulation,

navigation, natural language, safety, human robot interaction, and learning. While current researchers have partial solutions to many of these areas, integrating them all into a real-time system that is able to learn from instruction is a major challenge. Modeling the rules and uncertainty that govern the physical and social world is currently an unsolved problem. There are some robotic architectures and systems that focus on the person (e.g., ACT-R/E (Trafton et al. 2013), DIARC (Scheutz et al. 2007), but how to integrate those systems into functional AI systems is still very much an open question.

2. Symbol grounding and uncertainty: Robots sense and act in a highly uncertain world, which creates major challenges for endowing taskable systems with basic knowledge about the world. Such symbol grounding is critical to bridge high-level reasoning and taskability with low-level sensing and actuation. This inherent uncertainty takes a number of forms, primarily as: 1) perceptually grounding basic axioms about the world as symbols estimated from noisy sensor signals (such as for object recognition, Lawson and Trafton 2013), 2) predicting the effects resulting of actions for ensuring goals can be reached from executing actions in sequence (such as using physical simulation, Vondrak et al. 2012), and 3) estimating the intentions and goals of human teachers demonstrating tasks to robots (such as goal-directed learning from demonstration Chao et al. 2011).
3. Metrics: Another challenge is how to define metrics for evaluating an integrated interactive task learning system. Most such systems will be less computationally efficient than specific systems optimized for a specific task, so metrics and evaluations need to span across multiple tasks and domains.
4. Interoperability and time-sharing: Robotics has the ongoing problem that a particular algorithm often does not work on a different platform (sensor or robot). Science is, at its core, the ability to verify knowledge and findings independently through reproducible experimentation. In order to share knowledge, the challenge of the same running architecture across different physical platforms must be addressed. Further, we are within striking distance of having widespread tools for easy time-sharing of expensive state-of-the-art robots (such as the NASA Robonaut 2 and the Willow Garage PR2) over the Internet, using Robot Web Tools (Alexander et al. 2012) and the Robot Operating System (Quigley et al. 2009).

Possible Shared Challenge Problems:

1. Individual or collaborative assembly tasks (e.g., with Tinker toys).
2. Cleaning up toys in a kid's playroom (e.g., with unknown objects).
3. Preparing dishes (e.g., with new tools in unknown kitchen environments).
4. Preparing dishes in a common kitchen environment.
5. Taskable home care robot for an elderly or disabled individual, capable of cleaning up a home.
6. Performing tasks involving assembly and organization (preparing dining trays or drug regimens).

Personal Assistants
Many important kinds of intelligent agents today are purely software, providing services in the cyber world where so much of our thinking and work take place today. Intelligence analysts build models of countries, factions, parties, and people, while business analysts build models of products, markets, supply chains, and other aspects of their operating environment. Both types of analysts must deal with information that is incomplete, uncertain, and sometimes aimed at deception. Scientists, engineers, and doctors must handle an explosion of literature, within which there is reduplication, uncertainty, and much that is irrelevant. Software assistants need to be taskable and customizable to the needs of individual uses, and they need to learn new analytic practices and to rapidly get up to speed on new areas of interest without programming.

One of the major bottlenecks for such agents is deep natural language understanding. Progress in question-answering provides an interesting illustration. Accuracy for most question-answering systems has remained at roughly 30% for years, as the results from the TREC competition illustrated. By contrast, IBM's Watson achieved accuracy in the high 80s, using a combination of structured representations (including a 900 million element frame representation, automatically constructed via reading text) and statistical machine learning, which was enough to routinely beat human champions at the real-time TV quiz show Jeopardy! Watson's learning processes and reasoning processes were completely hand-tailored for the problem at hand, by a team of human experts who put in a person-century of work to succeed. Obviously, this approach does not scale. Interactive task learning could potentially enable anyone to customize and extend, in other words, to truly personalize, their software assistants.

The state of the art in such systems is still quite primitive. Learning by reading systems (e.g. Barker et al. 2007; Forbus et al 2007) can extract deep knowledge from texts. For example, Lockwood & Forbus (2009) describe a system that, after processing a textbook chapter expressed in simplified English and sketched diagrams, was capable of correctly answering 12 out of 15 questions from the end of the chapter. Salvucci (2014) has shown that knowledge from semantic web resources can be combined with ACT-R to provide answers to "factoid" questions. Project Halo has focused on exploring the kinds of reasoning needed to solve Advanced Placement test problems in multiple domains, while Forbus' Companion cognitive architecture has been used to learn to solve AP Physics problems (Klenk & Forbus 2009) and to learn to solve new physics problems via cross-domain analogies (Klenk & Forbus 2013). While progress continues to be made in all of these areas, the building of integrated systems that use them to experiment with interactive task learning could push the state of the art forward even faster, especially if improving the knowledge of such systems became an important class of experimental tasks.

Possible Shared Challenge Problems

1. Learning background knowledge. Using human-normed textbooks, including reading comprehension books, learn enough about the world to do well on human-normed tests, such as the New York Regents' exam.
2. Analysts Assistant. Given background materials on a new topic, build up a conceptual model of it, extending the model as new information becomes available, and responding to queries about it. Topics might include countries, crises, products, or research findings, while the background materials might include text, sketches, pictures, and video. Queries might include prediction questions, abductive questions (e.g. inference to best explanation, why might they be massing troops on the border?) and historical analogies.

Constructive Agents and Virtual Humans

We use "constructive agents" and "virtual humans" in this subsection to be inclusive and attempt to group several clusters of research and development that have emerged around closely related terminology that has somewhat different connotations across education, training, and entertainment applications. Constructive agents and virtual humans are implemented in software and run without the direct involvement or low-level, remote control from human operators. This distinguishes them from the "avatars" that represent human users in computer games and virtual world environments such as Second Life or OpenSim.

The term "virtual human" is more often used when the agent takes on some form of simulated physical embodiment and when they are designed to exhibit human-like behavior. An example might be a virtual character in a role-playing game that talks with the human game player. The term "constructive agent" is more common when there is no physical embodiment perceived by humans interacting with the agent and when human-like performance levels are not necessarily a constraint (e.g., it may be acceptable or even desirable for the agent to behave in a perfectly predictable and reliable manner every time it completes a task). An example of a constructive agent could be an agent that controls an airplane or tank in a simulation training application. These terms and associated underlying technologies have fuzzy boundaries, and the terminology is often used interchangeably across and within research communities.

Various forms of constructive agent and virtual human capability have been the focus of scientific and technical investments going back at least 30 years. Two summative National Research Council (NRC) reports (Pew & Mavor, 1998; Zacharias, MacMillan, & Van Hemel, 2008) have been published on the topic. Intended applications are most commonly in education and training, although they are increasingly commonplace in video games (e.g., Nareyek, 2007) and in film and television (e.g., Miles, 2004). Sub-types of these systems tend to cluster into opponents, teammates, and crowds.

Constructive agents or virtual humans are typically necessary to provide a realistic and interactive social environment in which human experience takes place in a virtual environment. However, these capabilities are costly to develop and, because they are often specialized during implementation for a particular application, they typically require significant additional manual refinement to adapt to a new application (even if very similar to the previous one). Consistent issues associated with brittleness and development costs have plagued research and application efforts. Even modest progress in the direction of ITL systems could result in a dramatic decrease in development costs and generalizability.

These challenges have motivated researchers to explore the use of machine learning to develop constructive agent and virtual human capabilities. For example, Laird and colleagues (van Lent, M. and Laird, J. E., 2001; Könik, T., & Laird, J., 2006) employ traces of entity behavior in simulation, along with annotated descriptions of goals and plans, to learn an agent behavior model. Automatic construction of interactive game characters has also been explored in the game and entertainment research community (e.g., Chang, et al, 2011). In almost all cases, there is little interaction (natural or otherwise) between the actual user of the virtual human, who may be in the best position to describe requirements, and the development process. A software developer acts as the intermediary, interpreting requirements and creating new behavior models from those requirements. This separation has led to the development of abstract programming interfaces, such as graphical programming tools and high-level languages, but these agents would be more useful, less brittle, and have a longer and more flexible life cycle if users could customize and specialize their behavior via natural interactions.

Learning behavior through natural interaction is being explored, however. For example, Permar and Magerko (2013) describe a system that learns to improvise a dance with a human partner. Although the scope of the learning interaction is limited to movements, the system readily creates new dance behaviors without being dependent on specialized programming or user training. Moving from dance partners to mission teammates, the Air Force Research Laboratory (AFRL) has been developing a Synthetic Teammate capability (Ball et al., 2010). It uses a theory of language processing (Ball, 2007, 2008) combined with a dynamically constructed situation model (Rodgers, Myers, Ball, & Freiman, 2012) to interpret and respond to text chat among teammates in simulated UAV missions. The Synthetic Teammate's relevance to the ITL objectives is in the natural, interactive manner in which it develops its internal mission situation model through text chat with teammates.

Interactive task learning could make a significant contribution toward several important challenges in constructive agents and virtual humans. First, conversational interactions that allowed an agent to gain insight into the specific goals and requirements of a particular user experience could be important for an agent that was primarily manually programmed. By analogy, imagine actors getting input about a scene or story arc from a director or a squadron of pilots receiving their mission brief. These conversations inform the players in carrying out their behavior in the scene and having greater context to make good decisions autonomously. Second, interactive task learning would help constructive agents and virtual humans migrate to new applications or even domains. An agent that knows how to fly a fighter plane should be able to learn to fly a commercial airliner or a cargo plan, for example.

Possible Shared Challenge Problems:

1. At-a-distance actors: Given guidance from a teacher, the agent would learn to exhibit/behave within given/prescribed norms. The challenge addresses the need for realistic "background characters" in film and television and in some kinds of simulation training. To address this challenge, the learning systems may need to "take in" cultural cues and to learn to exhibit them. Addressing this challenge likely requires embodiment (virtual human).
2. Platform operator: For this challenge, the agent would learn to control a simulated physical platform, such as a car, tank, or aircraft. Different vehicles and applications will have different requirements for physical fidelity and thus the complexity of control task will vary depending on the domain (ship at sea vs. supersonic aircraft). However, solutions likely share common properties and requirements that could be exploited across multiple research groups. This constructive agent approach probably does not require embodiment.
3. Role-players. For this challenge, assume some constructive agent or virtual human has been developed that exhibits high degree of capability and knowledge of in a domain and application. The challenge is to learn to be an effective role-player within that domain. That is, learn to customize how behavior is manifest in different situations based on the requirements for that context. This challenge has some potential pitfalls in terms of how the prior knowledge would need to be structured to support this capability. Further, it is a narrower problem than ITL is pursuing general, because it is focused on producing variation of behavior within a domain. However, it would offer immediate practical utility as a new feature for existing models/virtual actors. It might also be worthwhile as a goal of ITL research to frame ITL capability as one that could learn to exploit a virtual human or constructive agent, rather than necessarily being integrated within it.

Cognitive Science Research

Whereas recent AI approaches to task learning have primarily emphasized functionality and scope, recent Cognitive Science approaches have primarily focused on psychological plausibility and validity with respect to human learning. For example, a number of recent efforts have explored how computational models can, like people, learn by reading and following instructions (e.g., Anderson, Taatgen, & Byrne, 2005; Anderson et al., 2004; Salvucci, 2013; Taatgen, Huss, Dickison, & Anderson, 2008; Taatgen & Lee, 2003). In this work, the models encode instructions as declarative representations, and then proceduralize the instructions over time, thus moving from novice to expert performance. Lebiere et al (2013) extend that approach to building shared mental models in the domain of human-robot interaction and later apply it to sensemaking for geospatial intelligence analysis (Thomson et al, 2014). Those models combine instructions specifying a decision making procedure combined with learning to make individual decisions from demonstration and exploration.

Challenges:

Many challenges still remain in developing cognitive models of task learning. One of the most important challenges is moving beyond instruction following to other common forms of learning, such as learning by example; people are very adept at mixing forms of learning as they study new tasks, whereas current models, by and large, only follow a single method of learning. Another important challenge involves developing better models of natural language understanding; current cognitive science models have typically incorporated basic parsing and understanding, but only in a very limited scope, whereas interactive task learning and collaboration will require a much richer and more flexible model of understanding. Perhaps the most fundamental challenge is modeling not just the initial learning to perform the task from instructions and background knowledge but the entire learning curve including higher levels of expertise with the task which require the acquisition of high-level task concepts that are not directly instructable but instead must be discovered through experience.

Possible Shared Challenge Problems:

1. Simulated Students: Using a cognitive architecture as an underlying framework, one might develop simulated students that interactively receive instruction and, at the same time, practice new tasks and improve on these tasks through learning. Such a simulated student could serve to predict the difficulty of new problems, and could also be incorporated into intelligent tutoring systems for more accurate inference of a student's current skill level.
2. Interactive Learner for Psychological Experiments: Another potential challenge problem involves using a cognitive architecture to develop a model that can learn to perform psychological experiments in the way that humans do. In a standard experiment, a person enters with a large base of underlying procedural and declarative knowledge; during an instruction period and during the experimental task itself, the person augments this knowledge with an understanding of how to perform the specific task being asked. A preliminary model in this direction has been developed (Salvucci, 2013), but there remain significant challenges to making a broader interactive learner. First, the learner requires a significant natural-language component to accept instructions and translate them to actions; while natural-language understanding is critical to many areas of interactive task learning, its instantiation within a cognitive model is especially important in that it demonstrates how a cognitive architecture can transform the language to procedural knowledge within the architecture (e.g., production rules). Second, although there has been significant process on learning from instructions, the experiment learner needs to acquire knowledge in multiple ways — especially in learning by example — and these avenues remain largely unexplored within cognitive modeling. Third, the experiment learner's background knowledge is

extremely important and necessitates large-scale understanding and development of useful bodies of knowledge that can be applied to tasks; again, although some areas of such knowledge have been explored in more detail (e.g., mathematical knowledge), other areas remain less explored (e.g., spatial and graphical understanding) and offer avenues for further work.

3. Cognitive Debiaser: In contrast to physically embodied domains such as robotics, information manipulation in cyberspace is becoming an increasingly important part of both everyday life and professional activities. A potential challenge is to develop a system that could serve as a personal assistant to an intelligence analyst, modeling his cognitive processes at an individual level in order to alleviate some of their shortcomings such as cognitive biases and attentional bottlenecks. The task could be defined as receiving layers of information and issuing probability judgments over a space of hypotheses. Interactive Task Learning in that domain would include both developing the structure of decision making (e.g., the nature and order of available information, its semantics, the hypothesis space, etc.) and tailoring specific judgments to the capabilities and limitations of the individual analyst. This challenge would advance the state of the art in cognitive modeling by requiring the integration of a predictive, quantitative theory of the myriad of cognitive biases documented in the decision making literature. A potential application would be to evaluate the impact of structured analytic techniques (Heuer and Pherson, 2010) that have been proposed as a solution to cognitive biases. Metrics would be designed to quantify biases as deviations from rational normative performance.

Section 4. Characterizations of ITL Dimensions and their Variations

In this section, we attempt to characterize many of the dimensions of variability that will arise in ITL agents: what types of tasks can they learn, what types of knowledge can they learn, what types of interactions with a human can they support, and so on. A long-term goal is to develop agents that are able to support diversity in all of these dimensions, but comprehensive coverage is not necessary to build useful systems for the domains and applications described in Section 3. In the short-term, these dimensions help situate the contributions of a specific system by which subsets of these lists they cover, while in the long run, they provide stretch goals for achieving comprehensive interactive task learning systems.

1. Types of tasks

There appear to be many different types of tasks, or at least different ways of formulating tasks. Certain domains may have characteristic task formulations, such as many robot tasks involve achieving specific configurations of objects in 3D space; while at another extreme, some tasks for virtual agents involve constructing complex conceptual models of situations from multiple sources of information which might include uncertainty and deception. Our hypothesis is that for different types of tasks, different types of knowledge must be communicated to the agent, and that dimension is explored in detail below. In making these classifications, we have found it useful to distinguish between the external state of an environment and the internal information/knowledge structures in agents.

- Desired State Achievement. For these types of tasks, the purpose is to achieve a state of the world. In some cases, it might be possible to have a declarative description of the necessary and sufficient conditions for completing the task (such as the rules for winning chess); however, in some cases, the complexity of the desired state might be such that the agent can only recognize completion (such as write a good play or make a tasty dinner).
    - Environmental state: Achieve a specific configuration of objects in an environment, possibly constrained by additional rules on intermediate states or actions.
    - Knowledge/information state: Activities such as answer a question or perform a calculation. For example, an agent might learn how to find the cheapest fare for flights, or how to compute a pseudo random number.
- State Maintenance (homeostatic goals). Instead of achieving a state of the world, these tasks involve maintaining the state of the world within some constraints. Some of these can map directly onto classic control tasks, such as keeping a plane flying or juggling a set of balls. Homeostatic goals also occur in information domains. For example, there are commercial tools that require that an agent keep up with the news on a particular topic (e.g. sports scores) or monitor for product price changes and to keep a user informed of changes in information state.
- Procedures. Some tasks are best described as a set of actions to be performed as opposed to a state to be achieved or maintained. For example, homeostatic tasks such as patrolling a building are usually easier to describe as a set of waypoints or areas to visit, instead of the informational goal of maximizing situational awareness of an area. Other examples include human activities where the final state is not substantially different from the initial state, such as dancing, are hard to describe as state achievement or state maintenance.

2. Other important characteristics of tasks

- Satisficing vs. optimizing: is it sufficient to achieve a goal, or is there an additional constraint to maximize some value or minimize some cost function.
    - Single agent vs. multi-agent: cooperative vs. competitive
- Time limited vs. not

3. Types of knowledge: What an agent can learn about a task

- Task name: useful for human to refer to task in the future
- Task preconditions: what are situations in which the task can be pursued
- Task desirability conditions: when should the task be pursued
- Task definition (problem space)
    - State structure: relevant objects, concepts, categories, properties, and relations – those features that define the states in this task.

    - Legal & illegal states, constraints on states.
    - Initial state: the starting situation.
    - Goal: desired states.
    - Terminal states: independent of success or failure.
    - Legal & illegal actions, constraints on actions.
    - World/task dynamics that occur independent of the agent.
    - Procedures: some tasks are easier to define as procedures to execute rather than as states to be achieved.
- Supporting knowledge about components of task definition.
    - In general, the expectation is that learning the components of task definition that are listed above is a simple activity; however, any of these could be complex enough to require learning a new task, such as learning how to monitor the current state to determine if a task constraint has been violated.
    - Subtask structures: a decomposition of the current task into subtasks that can be learned.
    - Action models: knowledge about how the agent's actions change the state of the environment.
- Strategic Information (how to perform the task well). Learning this type of knowledge does not distinguish ITL systems from many existing learning systems, which acquire this knowledge through either self-directed exploration (such as reinforcement learning) or guidance from a teacher.
    - Important state features
    - State evaluations.
    - Operator-state evaluations
- Explanations as to why the task is structured as it is

4. Types of Interaction

An instruction can use many different modalities for communicating with the agent. Below we list the most common as well as discuss other dimensions of variation in the interaction between an agent and teacher.

- Modality of knowledge specification:
    - Language
    - Gestures
    - Sketches
    - Demonstration via manipulation of the robot, including remote control
    - Demonstration where another agent performs the desired action and the agent must interpret the activity and map it onto its own behavioral capabilities. Sometimes described as learning by observation.
    - Feedback/reward, the teacher provides only a value that represents the quality of the agent's behavior.

- Naturalness of language (or other modalities) – spectrum from natural to artificial
    - Free-form natural language vs. restricted natural language vs. artificial language
- Initiative in interaction
    - Teacher-driven
    - Student-driven
    - Mixed initiative
    - Not interactive – complete specification without any interaction
- Grain-size of interaction
    - Fine-grain. Each interaction is a small number of concepts, modulated by interaction.
    - Medium-grain. Sets of related concepts are specified together.
    - Coarse-grain. The complete task specification is provided at once, such as in written instructions. This minimizes interaction.
- Types of environment in which objects/concepts/relations/actions/… are referenced
    - Shared physical. Allows the teacher to point at objects and refer to them using spatial relations.
    - Hypothetical environment.
    - Earlier in the interaction, so that the participants can use
- When interaction happens
    - During task execution.
    - Before task execution
    - After task execution

5. Characteristics of Agents

This item characterizes variants in the capabilities of ITL agents, distinguishing different types of pre-encoded knowledge, and different cognitive capabilities that can aid in task learning and performance.

- Pre-encoded knowledge of environment used to learn new task knowledge
    - Objects, concepts, categories, properties, relations
    - Spatial structures
    - Primitive actions
    - Language capabilities
    - Knowledge of dialog
- Additional cognitive capabilities that enhance task learning and performance
    - Reasoning, decision making, and planning abilities so that agent can make additional deductions to support extract of knowledge from instruction and the environment.
    - Active exploration of environment and task.
    - Other learning capabilities, such as reinforcement learning, strategy learning, concept acquisition, clustering, …

    - Ability to detect knowledge inconsistencies.
    - Ability to correct knowledge inconsistencies.
    - Ability to explain reasons for its behavior.
    - Ability to interact with other agents.
    - Ability to model the cognitive processes and behavior of other agents.

6. Characteristics of Environment

An ITL agent learns tasks within an environment. In this item, we highlight different dimensions of tasks that could have an impact on interactive task learning.

- Overall environmental characteristics
    - Internal problems. These are tasks the agent performs without interacting with an external environment. These can be internal models of an external environment (imagine playing a game of Tic-Tac-Toe), but can also include internal calculations that involve reasoning and memory access.
    - External software systems. Here the agent is interacting with a piece of software.
    - Simulation of real-world environments.
    - Real-world robotic environments.
- Fully vs. partially observable environment. Can the agent perceive all aspects of its environment all of the time? This is true in many games, but is rarely true in other tasks.
- Propositional vs. relational features. Can the task state be represented as a feature vector of fixed length (propositional), or does it require relational features that vary based on the environment (such as when there are different numbers of objects in the world). Propositional representations are much easier to represent, reason over, and learn over.
- Discrete vs. continuous properties (cell/grid-based worlds vs. real-world)
- Discrete vs. continuous actions (in terms of agent control)
- Deterministic vs. non-deterministic actions
- Reversibility of actions.
- Dynamism of environment. One where change happens only through agent actions vs. one where dynamics are not completely under the control of the agent (such as because of other agents).

7. Characteristics of Teachers

- Task expertise
    - Confidence in their expertise
- Instruction expertise
    - Ability to create common ground with student – communicate important aspects, instructional focus, …
- Preferred and types of modality of interaction
- Commitment to teaching
- Reliability of instructions

- Trustworthiness
- Engagement & interactiveness & …
- Emotional engagement
- Single vs. multiple teachers
- Teacher is not a teacher but another participant
- Shared embodiment in the environment with the student vs. remote instruction where teacher cannot sense the environment

Section 5. Community Building for Future Research in ITL

Below are the results of discussion on community building, divided into existing resources and then planned activities. An additional important community building activity would be to create repository of tasks, instructions, data, and performance benchmarks, which could speed up progress, develop critical mass and facilitate convergence in the field.

Resources

1. Knowledge bases that could be useful for providing an ITL with basic knowledge of the world, eliminating the need for the teacher to provide *all* knowledge about a task.
    a. Cyc: http://www.cyc.com/
        i. OpenCyc http://www.cyc.com/platform/opencyc
        ii. ResearchCyc http://www.cyc.com/platform/researchcyc
    b. KnowRob: http://www.knowrob.org/
    c. WordNet: http://wordnet.princeton.edu/
    d. RoboEarth: http://roboearth.org/
    e. NULEX: http://www.qrg.northwestern.edu/resources/nulex.html
    f. Framenet: https://framenet.icsi.berkeley.edu/fndrupal/
    g. Yago: http://www.mpi-inf.mpg.de/yago-naga/yago/
    "YAGO2s is a huge semantic knowledge base (10M entities, 120M facts), derived from Wikipedia WordNet and GeoNames."
    h. DBpedia: http://dbpedia.org/About "A crowd-sourced community effort to extract structured information from Wikipedia and make this information available on the Web."
    i. Wiktionary: http://en.wiktionary.org/ A free online dictionary that designed to be the lexical companion to Wikipedia.
    j. WikiHow: http://www.wikihow.com/ A guide to humans for how to do various tasks. Uses combinations of text, figures, and photographs. Not interactive and assumes lots of background knowledge.
2. Simulators
    a. Gazebo http://gazebosim.org/
3. Available from specific labs
    a. Kenneth Forbus
        i. NuLex

        ii. OpenCyc flat files http://www.qrg.northwestern.edu/OpenCyc/opencyc_flat_files.htm
        iii. Minimal OpenCyc ontology (< 1mb)
        iv. FreeCiv interface
        v. Case Mapper (SME and MACFAC)
            1. Binary - KQML interface
            2. http://www.qrg.northwestern.edu/software/casemapper/index.html
        vi. CogSketch http://www.qrg.northwestern.edu/software/cogsketch/
    b. Matthias Schuetz
        i. Instruction data set for a human-human remote search task. (not a learning dataset)
        ii. Instructions on how to perform simple tasks like set the table (a student has it, he wasn't sure if she would like to share)
    c. Ken Koedinger
        i. Datashop: curricula for students learning with a tutor: https://pslcdatashop.web.cmu.edu
    d. John Anderson
        i. ACT-R: http://act-r.psy.cmu.edu
    e. John Laird
        i. Soar: http://sitemaker.umich.edu/soar/home

Community Building

During the meeting, we spent time discussing next steps for community building. Although we discussed many options, below are the next steps we have planned:

1. *Workshop Report.* Finish this report and distribute to researchers who might be interested in joining our community.
2. *Magazine Article.* Using this report as material, write an article on interactive task learning and submit it to AI Magazine in August 2014 for publication in Spring 2015.
3. *Journal Special Issue.* At some point in the future, we might consider a special issue of research papers in a journal, such as HRI journal, or Topics in Cognitive Science.
4. *Workshops.* We are going to investigate options for holding additional workshops.
    a. A follow-up NSF workshop that would expand the people invited and develop more of a strategic plan, including a document that addresses the Heilmeier Catechism (from DARPA).
    b. Struengmann Forum http://www.esforum.de/ This would be a chance for international engagement. There is a long lead time to organize one of these, so we are starting to pursue it immediately. (Kevin Gluck)
    c. National Academy of Science (2 day workshop).
    d. German Dagstuhl Seminars http://www.dagstuhl.de/en/program/dagstuhl-seminars

e. AAAI Fall or Spring Symposium. AAAI Workshops. No definitive plans yet to organize one, but a possibility when we want to have broader engagement.
f. Advances in Cognitive Systems. At this conference, we could organize a co-located one day workshop.

5. *Conference Panels.* We will consider organizing panels on ITL for existing conferences and workshops.
   a. AAAI spring symposium
   b. HRI, ICCM
6. *Conferences*. (most appropriate in bold)
   a. Robotics: **HRI**, ICRA, IROS, RSS
   b. Cognitive Science: **ICCM**, Cognitive Science, ICDL, BRMS, CHI
   c. AI: **ACS,** AAAI, AAMAS (learning by demonstration)
   d. Computer Games: AIIDE (directable characters)
7. *Other researchers to involve in future discussions of ITL.*
   a. Michael Beetz (Technische Universität Muenchen) Robotics
   b. Michael Black (Max Planck) Perception
   c. Maya Cakmak (UW) HRI
   d. Joyce Chai (MSU) HRI
   e. Sonia Chernova (WPI) HRI/Robotics
   f. Hiroshi Ishiguro (Osaka) Robotics
   g. Ken Koedinger (CMU) Education and HCI
   h. Maya Mataric (USC) HRI/Robotics
   i. Charles Rich (WPI) Task Learning
   j. Neils Taatgen (Groningen) Cognitive Modeling
   k. Manuela Veloso (CMU) HRI/Robotics

Funding Opportunities

We also had discussions about different funding opportunities, including NSF and DoD

APPENDIX I: WORKSHOP SCHEDULE

University of Michigan, May 13-14, 2014

May 13	**Topic**
8:30-10:00am	Introductions: ~5 minute presentations by participants on interest in task learning. • Includes example research and systems, and current research projects.
10:00-noon	Task learning definition discussion: • What are different overall approaches to task learning, their strengths, and their weaknesses? • What knowledge is communicated, how is it communicated, what interactions are there between teacher and learner?
Noon-1pm	Lunch: Informal discussions of morning topics.
1pm-3pm	Possible application areas for task learning: ~5 min. presentations • Possibilities: games, personal assistants, psych. experiment modeling, virtual training, homecare robots, military robotics o Ken Forbus & John Laird: Games and puzzles o John Anderson, Dario Salvucci, Christian Lebiere: Psych. experiment modeling o Kevin Gluck & Bob Wray: Virtual training systems o Chad Jenkins, Matthias Scheutz, Andrea Thomaz & Greg Trafton: Collaborative robots • Why is this an important application area? • Some characterization of the tasks in this area along dimensions discussed above.
	•
3pm-5pm	Identify and discuss the science and technology issues in task learning: • Possibilities: natural language, gesture recognition, HRI, language learning, skill learning, proceduralization of declarative knowledge, analogy, using background knowledge, knowledge transfer, explanation, spatial and temporal reasoning, … • **Note: because of time constraints, this topic was not discussed at the workshop.**
7-9pm	Dinner (cash bar): Mediterrano, 2900 S. State St, Ann Arbor, MI 48108
May 14	
9am-10am	Possible metrics for evaluating task learning systems. • What claims will we want to make? • What data can help us evaluate whether our approaches support those claims?
10am-noon	What materials can be shared across research groups? • Those that already exist - how-to instructions and videos? Related research on task learning from other fields. • New materials and data that need to be created.
Noon-1pm	Lunch: Informal discussions of earlier topics or next topic.
1pm-3pm	Strategize about establishing and growing a research community on task learning • Follow on workshop or AAAI symposia? • How can we secure funding to support research in task learning?
3pm	Head to DTW for flights. You are also welcome to stay around and schmooze for a while.

APPENDIX II: ANNOTATED BIBLIOGRAPHY

Ordered chronologically within each section.

Task learning from language.

1. Simon, H. A., & Hayes, J. R. (1976). The understanding process: Problem isomorphs. Cognitive Psychology, 8, 165-190.
   Simon, H.A. (1977). Artificial intelligence systems that understand. Proceedings of the Fifth International Joint Conference on Artificial Intelligence, 2, 1059-1073.
   UNDERSTAND was the first implemented task learning system (the papers don't describe a complete implementation – lots of hand simulation, so its implementation status is not completely clear). It could learn isomorphs of the Tower of Hanoi puzzle from written descriptions (not interactive). Their research showed that the way the task was described influenced a human's internal representation of the task state, which in turn influenced problem solving efficiency/skill.
2. Rychener, M. D. and Newell, A., (1977). An instructable production system: Basic design issues, in D. A. Waterman and F. Hayes-Roth, Eds., Pattern-Directed Inference Systems, New York, NY: Academic Press.
   Rychener, M. (1980). Approaches to Knowledge Acquisition: The Instructable Production System Project. In Proceedings of the First National Conference on Artificial Intelligence, 228–230. Menlo Park, Calif.: American Association for Artificial Intelligence.
   Rychener, M. (1983). The Instructable Production System: A Retrospective Analysis. In *Machine Learning: An Artificial Intelligence Approach*, eds. R. Michalski, J. Carbonell, and T. Mitchell, 429–459. San Mateo, Calif.: Morgan Kaufmann.
   IPS: The purpose of this project was to use instruction to teach a system instead of programming. The project was not successful, but the research issues encountered (the need for a formulation of task structure – the problem space computational model) led to the development of Soar.
3. Crangle C. and Suppes P. (1994). Language and Learning for Robots, CSLI Lecture notes No. 41, Centre for the Study of Language and Communication, Stanford, CA. ISBN:1881526194
   "Robot technology will find wide-scale use only when a robotic device can be given commands and taught new tasks in a natural language. This text pursues a theoretical approach to language and learning in developing the notion of an instructable robot that derives intelligence from human interaction. … The authors also examine the use of explicit verbal instruction to teach a robot new procedures; propose ways a robot can learn from corrective commands containing qualitative expressions, and discuss the machine-learning of a natural language use to instruct a robot in the performance of simple physical tasks."
4. Huffman, S. B., and Laird. J. E. (1995). Flexibly Instructable Agents. Journal of Artificial Intelligence Research, 3, 271-324.
   Instructo-Soar. Through restricted natural language, it learns new tasks and subtasks in Soar that can be built up hierarchically, such as picking up and stacking blocks in a simulated robotic environment. Shows transfer in learning among tasks. "Instructo-Soar meets three key requirements of flexible instructability that distinguish it from previous systems: (1) it can take known or unknown commands at any instruction point; (2) it can handle instructions that apply to either its current situation or to a hypothetical situation specified in language (as in, for instance, conditional instructions); and (3) it can learn, from instructions, each class of knowledge it uses to perform tasks."

5. Lauria, S., Bugmann, G., Kyriacou, T., Bos, J. and Klein, E. (2001). Training personal robots using natural language instructions. IEEE Intelligent Systems, 16:38-45, 2001.
   Lauria, S., Bugmann, G., Kyriacou, T. and Klein, E. (2002). Mobile robot programming using natural language, Robotics and Autonomous Systems 38 (3-4): 171- 181
   They propose Instruction-Based Learning (IBL). Via speech, a human instructor can define a new procedure that is a composition of existing primitives, and that task can then be executed in the future. This was done in a route following task with a small robot navigating in a miniature village. It starts with sophisticate primitive capabilities, that include visual search and planning. The exact status of its capabilities (and implementation) aren't clear from the second paper.
6. Dominey, P.F., Mallet, A., Yoshida, E. "Real-time cooperative behavior acquisition by a humanoid apprentice", Humanoid Robots, (2007) 7th IEEE-RAS International Conference on Humanoid Robotics, 270–275.
   Dominey PF, Mallet A, Yoshida E. (2007). Progress in Programming the HRP-2 Humanoid Using spoken Language, Proceedings of ICRA 2007, Rome.
   Lallee S, Yoshida E, Mallet A, Nori F, Natale L, Metta G, Warneken F, Dominey P.F. (2010) Human-Robot Cooperation Based on Learning & Spoken Language Interaction From Motor Learning to Interaction Learning in Robots, Studies in Computational Intelligence, vol. 264, Springer-Verlag
   Supports using restricted language to specify a robot's behavior, and these behaviors can be saved away as macros for later reuse.
7. Anderson, J. (2007). *How Can the Human Mind Occur in the Physical Universe?* Oxford University Press.
   Includes an ACT-R model of task acquisition. This was in the context of creating cognitive models where the researcher does not build the model – the system does. In the same spirit as UNDERSTAND – addressing the issue of how procedural knowledge can be acquired through problem solving experience guided by declarative instructions in memory. Used for only a limited number of problems, including the Pyramid Problem. Not interactive task specification, but batch.
8. Allen, J., Chambers, N., Ferguson, G., Galescu, L., Jung, H., Swift, M., and Taysom, W. (2007). PLOW: A Collaborative Task Learning Agent. In Proc. Conference on Artificial Intelligence (AAAI) . Vancouver, Canada.
   A collaborative task learning agent that acquires procedural knowledge through a collaborative session of demonstration, learning, and dialog. The human teacher provides a set of tutorial instructions accompanied with related demonstrations using which the agent acquires new procedural knowledge. Although the learning is human-demonstration driven, the agent controls certain aspects of its learning by making generalizations without requiring the human to provide a large number of examples.
9. Rybski, P., Stolarz, J., Yoon, K., and Veloso, M. (2008). Using dialog and human observations to dictate tasks to a learning robot assistant. Journal of Intelligent Service Robots, Special Issue on Multidisciplinary Collaboration for Socially Assistive Robotics, 1(2):159-167.
   Mericli, C., Klee, S., Paparian, J., Veloso, M. (2013). An Interactive Approach for Situated Task Teaching through Verbal Instructions. **AAAI Workshops**. Follow on publication in AAMAS 2014.
   Allows specification of behavior through language – essentially programming with language over a set of primitive, pre-encoded action that a robot can then execute in the future. An example instruction: *when I say deliver message, If Person1 is present, Give message to Person1....*

10. Mohan, S., Mininger, A., Kirk, J., and Laird, J. E. (2012). Acquiring Grounded Representation of Words with Situated Interactive Instruction. Advances in Cognitive Systems.

    Mohan, S. and Laird, J. E. (2014) Learning Goal-Oriented Hierarchical Tasks from Situated Interactive Instruction, AAAI 2014 (to appear).

    ROSIE: An agent implemented in Soar that can interactively acquire of *basic* concepts such as attributes of objects (color: *red*, size: *large*), spatial relationships (*right-of*), and simple actions (*move*) through language interaction in a real-world robotic environment. Rosie also acquires new vocabulary of adjectives, nouns, prepositions, and verbs that are grounded in basic concepts and can be used in interactions. It can learn hierarchical verbs that are compositions of its primitives. The learned knowledge is first stored in semantic memory, but procedural knowledge is learned by converting semantic knowledge to rules through chunking over retrospective analysis of the task.

11. Salvucci, D. D. (2013). Integration and reuse in cognitive skill acquisition. Cognitive Science 37. 829-860.

    An ACT-R system that focuses on the integration and reuse of previous skill knowledge and the proceduralization of this knowledge. Achieves competent behavior in many diverse tasks, but the commands are limited to a restrictive syntax that specifies policy behavior rather than general task and goal knowledge. Not interactive task specification, but batch.

    "…,the paper proposes a computational model of skill acquisition from instructions focused on integration and reuse, and applies this model to account for behavior across seven task domains."

12. Kirk, J. and Laird J. E. (2013). Learning Task Formulations through Situated Interactive Instruction. Proceedings of the 2nd Conference on Advances in Cognitive Systems. Baltimore, Maryland.

    ROSIE: Follow on research to Mohan et al (2012). Here the emphasis is on learning simple games and puzzles. The system learns the goal states, restrictions on actions, failure states, etc. within the context of a physical robot that first learns the game from language and then plays it by controlling the robot.

13. Hinrich, T., and Forbus, K. (2013). X Goes First: Teaching Simple Games through Multimodal Interaction. Proceedings of the Second Conference on Advances in Cognitive Systems.

    Uses CogSketch to teach an agent developed in the Companions architecture how to play simple games like Tic-Tac-Toe and Hexapawn. The system incrementally creates a GDL description of the task from an interaction with a user that includes instruction and demonstration. The GDL specification is then interpreted so that a Companions agent can play the game.

14. Petit, M., Lallee, S., Boucher, J.-D., Pointeau, G., Cheminade, P., Ognibene, D., Chinellato, E., Pattacini, U., Gori, I., Martinez-Hernandez, U., Barron-Gonzalez, H., Inderbitzin, M., Luvizotto, A., Vouloutsi, V., Demiris, Y., Metta, G., Dominey, P.F. (2013). The Coordinating Role of Language in Real-Time Multimodal Learning of Cooperative Tasks, IEEE Transactions on Autonomous Mental Development, 3-17 5(1)

    "These requirements include the ability to negotiate a shared plan using spoken language, to learn new component actions within that plan, based on visual observation and kinesthetic demonstration, and finally to coordinate all of these functions in real time. We present a cognitive system that implements these requirements, and demonstrate the system's ability to allow a Nao humanoid robot to learn a nontrivial cooperative task in real-time."

Task modification, optimization, or extension through language.

1. McCarthy, J. (1958). Programs with common sense, Symposium on Mechanization of Thought Processes. National Physical Laboratory, Teddington, England.
   Advice Taker: Lays out the idea of a system that learns from interaction with a human. Did not emphasize task acquisition per se, but emphasized advice to improve performance.
2. Winograd, T. (1972). Understanding Natural Language, New York: Academic Press
   SHRDLU did language understanding in a simulated blocks world. It did some learning as the user could define compositions of blocks (such as a tower) that the system would remember and could construct and answer questions about. It did not learn completely new tasks.
3. Hayes-Roth, F., Klahr, P. and Mostow, D. J. "Advice taking and knowledge refinement. An iterative view of skill acquisition " in J. A. Anderson (ed.). Cognitive Skills and their Acquisition, 231-253. Erlbaum. 1981.
   Provides an overview of the process of advice taking, using the game of Hearts as an example. Analyzes the types of knowledge required in the task and describes systems to perform different phases of interpreting instructions.
4. Mostow, D. J., "Learning by being told: Machine transformation of advice into a heuristic search procedure," in J. G. Carbonell. R. S. Michalski, and T. M. Mitchell (eds.), Machine Learning. Palo Alto, CA: Tioga Publishing Company, 1982.
   Describes BAR, a system that operationalizes advice for the game of Hearts. It could operationalize three different pieces of advice, each of which represented a different challenge in terms of not being easy to operationalize. "… taking advice that is non-operational, i.e. expressed in terms of data or actions unavailable to the machine, and transforming it into a procedure executable using only the available operations This process is called operationalization."
5. Blythe, J. (2005). Task Learning by Instruction in Tailor, *In Proceedings of Conference on Intelligent User Interfaces (IUI05).* San Diego
   "We introduce Tailor, a system that allows users to modify task information through instruction. In this approach, the user enters a short sentence to describe the desired change. The system maps the sentence into valid, plausible modifications and checks for unexpected side-effects they may have, working interactively with the user throughout the process." Emphasizes task modification, and assumes a user interface that allows the user to see the task specification and how the modification changes it.
6. Klenk, M. and Forbus, K. (2009) Analogical model formulation for transfer learning in AP Physics. *Artificial Intelligence*, 173(18):1615-1638.
   Describes how structure-mapping can be used for within-domain transfer to learn how to do AP Physics problems, using representations constructed by the Educational Testing Service, the organization that administers AP exams to high school students in the United States.
7. Weitzenfeld, A., Ejnioui, A. and Dominey, P. (2010). Human robot interaction: Coaching to play soccer via spoken-language. In IEEE/RAS Humanoids'10 Workshop on Humanoid Robots Learning from Human Interaction.
   "To demonstrate the interaction model we described how to coach a robot to play soccer by teaching new behaviors at two levels: (i) individual basic behaviors trained from a sequence of existing actions and interrogations, and (ii) hierarchical multi-robot strategies trained from newly trained sequences." Looks like it uses very constrained language to specify rules. Not "natural" and the scope of what can be specified seems limited. Maybe closer to task specification languages than task learning.

8. Hinrichs, T., and Forbus, K. (2011) Transfer learning through analogy in games. *AI Magazine*, 32(1), 72-83.
   Describes how Companions can learn simple tile-based games, and speed the learning of new games via analogical transfer of knowledge at multiple levels. The number of cross-domain problems, developed by others, is larger than any prior published result.
9. Klenk, M., Forbus, K., Tomai, E., and Kim, H. (2011) Using analogical model formulation with sketches to solve Bennett Mechanical Comprehension Test problems. *Journal of Experimental and Theoretical Artificial Intelligence*, 23(3):299-327.
   Describes how Companions can use causal explanations tied to sketches of specific physical systems to learn to solve the kinds of problems found on a commonly used test of spatial ability and mechanical reasoning.
10. Cantrell, R., Talamadupula, K., Schermerhorn, P., Benton, J., Kambhampati, S., & Scheutz, M. (2012). Tell me when and why to do it!: Run-time planner model updates via natural language instruction. In Proceedings of the 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI-12) (471–478). Boston, USA: ACM.
    A mobile robotic system implemented in DIARC that can be taught individual commands via language commands by specifying preconditions ("you are at a closed door"), action definitions ("you push it one meter"), and postconditions ("you will be in the room"). Does not seem to have the ability to learn completely new tasks that have complex internal structure.
11. Klenk, M. and Forbus, K. (2013) Exploiting persistent mappings in cross-domain analogical learning of physical domains. *Artificial Intelligence*, 195:398-417. 2013.
    Describes how new physical domains can be learned by using analogies between worked solutions. Between-domain correspondences are accumulated to facilitate distant transfer.
12. McFate, C., Forbus, K., & Hinrichs, T. (2014) Using narrative function to extract qualitative information from natural language texts. *Proceedings of AAAI 2014*.
    Shows how narrative function, a higher-level language construct, can be used to robustly construct qualitative models from text. One experiment shows that type-level qualitative models can capture a common form of advice given in strategy games, and lead to performance improvements.

Learning from demonstration or observation systems.

Most learn procedural knowledge for selecting actions based on either observing a human, or from being controlled by a human. There are many such systems, and I've only included a subset. For example, there are many systems that I have not included that learn non-hierarchical policies using reinforcement learning by observing a human, and some that use inverse reinforcement learning (they learn the reward function).

1. van Lent, M. and Laird, J. E. (2001). Learning procedural knowledge through observation. In *Proceedings of the 1st international conference on Knowledge capture* (K-CAP '01). ACM, New York, NY, USA, 179-186.
   A system that learns hierarchical procedural knowledge for tasks by observing human behavior in a simulator. The behavioral trace is annotated with goal structures. Applied to learning tactics for simulated tactical aircraft.
2. Nicolescu, M. N. and Mataric, M. J. (2003). Natural methods for robot task learning: Instructive demonstrations, generalization and practice. In Proc. of AAMAS.

They propose an approach for learning that consists of instructive demonstration, generalization over multiple demonstrations, followed by practice. Language is used to refine/clarify what is learned by demonstration. "Our goal is to develop a flexible mechanism that allows a robot to learn and refine representations of high level tasks, from interaction with a human teacher, based on a set of underlying capabilities (behaviors) already available to the robot." Restricted to learning FSM-style representations of tasks.

3. Bentivegna, D., Atkeson, C., and Gordon, C. (2004). Learning Similar Tasks from Observation and Practice. Robotics and Autonomous Systems 47:163–169.
   A framework that learns subgoals from segmented observed data. The system can use its experience to optimize its policy of selecting subgoals. Does not learn new tasks or subtasks but learns selection knowledge (a policy) to improve performance on existing tasks with known actions.
4. Könik, T., & Laird, J. (2006). Learning goal hierarchies from structured observations and expert annotations, Machine Learning 64 (1-3), 263-287.
   "We describe a relational learning by observation framework that automatically creates cognitive agent programs that model expert task performance in complex dynamic domains. Our framework uses observed behavior and goal annotations of an expert as the primary input, interprets them in the context of background knowledge, and returns an agent program that behaves similar to the expert."
5. Argall, B. D., Chernova, S., Veloso, M., and Browning, B. (2009). A Survey of Robot Learning from Demonstration. Robotics and Autonomous Systems, 57(5):469-483.
   Surveys learning from demonstration approaches. These approaches do not focus on task learning, but emphasize learning control policies for low-level motor control.
6. Grollman, D., and Jenkins, O. (2009). Can We Learn Finite State Machine Robot Controllers from Interactive Demonstration? In *From Motor Learning to Interaction Learning in Robots*: Studies in Computational Intelligence 405-429.
   This system formulates the problem of task acquisition as inferring a finite-state machine from segmented observations of a demonstrator performing the task.
7. Barbu, A., Narayanaswamy, S., and Siskind, J. M. (2010). Learning physically-instantiated game play through visual observation. In Proc. of ICRA'10, 1879–1886.
   A specially engineered robotic system that learns to play simple 3x3 board games, like Tic-Tac-Toe and Hexapawn, by observing random legal game play between two other agents. Customized for specific spatial configurations.
8. Chao, C., Cakmak, M., and Thomaz, A.L. (2011). Towards grounding concepts for transfer in goal learning from demonstration. In Proceedings of the Joint IEEE International Conference on Development and Learning and on Epigenetic Robotics (ICDL-EpiRob).
   Integrated task learning of task goals from demonstration.
9. Kaiser, Ł. (2012). Learning Games from Videos Guided by Descriptive Complexity. In Proceedings of the 26th Conference on Artificial Intelligence, AAAI-12, pp. 963–970. AAAI Press.
   Learns some board games through visual observation. The state is represent with relational structures, but these structures are predefined, namely rows, columns, and diagonals in the board grid.
10. Hinrichs, T., & Forbus, K. (2012) Learning qualitative models via demonstration. *Proceedings of AAAI 2012.*
    Learns qualitative models by observing the actions of an instructor in a game (Freeciv), and using those causal models subsequently to improve its own performance in a task within the game.

Task specification systems (not integrated into an agent at runtime, but involve separate compilation steps):

1. Yost, G. (1993). Acquiring knowledge in Soar. *IEEE Expert, 8*(3), 26-34.
   TAQL: A language of specifying tasks using the Problem Space Computational Model (PSCM). It compiled into Soar.
2. Genesereth, M., and Love, N. 2005. General game playing: Game description language specification. Technical report, Computer Science Department, Stanford University, Stanford, CA, USA.
   GDL: A formal language for specifying games – has its roots in PROLOG. Not meant for informal human communication. Programs take GDL as input and create programs that can play the games. Emphasis is on competence. Efficiency is important because many of the programs use search techniques, and the faster they can run, the deeper they can search.
3. Jones, R. M., Crossman, J. A., Lebiere, C., & Best, B. J. (2006). An abstract language for cognitive modeling. In *Proceedings of the Seventh International Conference on Cognitive Modeling,* 160-165. Trieste, Italy.
   HLSR: A High-Level Specification Representation. Not formally based on PSCM, but had similar structure. Was compiled into both Soar and ACT-R.
4. Cohen, M. A., Ritter, F. E., & Haynes, S. R. (2010). *Applying software engineering to agent development*. AI Magazine. 31(2), 25-44.
   HERBAL: Another language for specifying tasks using the PSCM. It also compiled into Soar.
   This paper includes an extensive review of task specification languages.
5. Langley, P. Trivedi, N. and Banister, M. (2010). A command language for taskable virtual agents. *Sixth Conference Artificial Intelligence and Interactive Digital Entertainment*. Stanford, Ca, AAAI Press.
   An instructional command language that is compiled into Icarus. Includes constructs such as while loops and if statements.

Systems that learn mappings between language and task, but don't learn new tasks.

1. Tellex, S., Kollar, T., Dickerson, S., & Walter, M. (2011). Understanding natural language commands for robotic navigation and mobile manipulation. Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (pp. 1507–1514). San Francisco, CA: AAAI Press.
2. Chen, D., & Mooney, R. (2011). Learning to interpret natural language navigation instructions from observations. Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (pp. 859–865). San Francisco, CA: AAAI Press.
   A unified agent architecture for human robot collaboration that combines natural language processing and common sense reasoning. A planning agent that relies on communication with the human to acquire further information about underspecified tasks. Does not do task learning, but learns mappings from language to existing navigation tasks.

Learning by Reading Systems

1. Barker, K., Agashe, B., Chaw, S., Fan, J., Friedland, N. Glass, M., Hobbs, J., Hovy, E., Israel, D., Kim, D., Mulkar-Mehta, R., Patwardhan, S., Porter, B., Tecuci, D., Yeh, P. (2007) Learning by Reading: A prototype system, performance baseline, and lessons learned. *Proceedings of AAAI-2007*
   Describes the MOBIUS system, which processed single paragraphs and constructed frame

representations from text.. Attempted to handle a wide range of syntactic variation, but only on the topic on how hearts work.

2. Forbus, K., Riesbeck, C., Birnbaum, L., Livingston, K., Sharma, A., & Ureel, L. Integrating natural language, knowledge representation and reasoning, and analogical processing to learn by reading. *Proceedings of AAAI-2007*.
   Describes the Learning Reader system, which processes short stories expressed using simplified English syntax, concerning world history. Introduced the idea of *rumination*, where a system improves its understanding by asking itself questions off-line.
3. Lockwood, K. and Forbus, K. (2009). Multimodal knowledge capture from text and diagrams. *Proceedings of KCAP-2009*
   Demonstrated that Mayer's multimodal learning theory could be implemented via analogical mapping between representations produced by independently understanding a text and and associated diagram. Used semi-automatic processing of simplified English text plus sketches to build up a model of the first chapter of a book on basic machines.

APPENDIX III: ADDITIONAL REFERENCES

1. B. Alexander, K. Hsiao, O. C. Jenkins, J. Lee, B. Suay, and R. Toris, "Robot Web Tools [ROS Topics]," IEEE Robotics & Automation Magazine, 19(4) 20-23, 2012.
2. Ball, J., Myers, C. W., Heiberg, A., Cooke, N. J., Matessa, M., Freiman, M. & Rodgers, S. (2010). The synthetic teammate project. *Computational & Mathematical Organization Theory*, *16*(3), 271-299. (DOI: 10.1007/s10588-010-9065-3)
3. Ball, J. (2007). A bi-polar theory of nominal and clause structure and function. *Annual Review of Cognitive Linguistics*, (pp. 27-54). Amsterdam: John Benjamins.
4. Ball, J. (2008). A naturalistic, functional approach to modeling language comprehension. *Papers from the AAAI Fall 2008 Symposium, Naturally Inspired Artificial Intelligence*. Menlo Park, CA: AAAI Press.
5. M. Cakmak, C. Chao, and A.L. Thomaz, "Designing Interactions for Robot Active Learners." in IEEE Transactions on Autonomous Mental Development, 2010.
6. M. Cakmak and A.L. Thomaz, "Designing Robot Learners that Ask Good Questions." In Proceedings of the International Conference on Human-Robot Interaction (HRI), 2012.
7. C. Chao, M. Cakmak and A.L. Thomaz, "Towards Grounding Concepts for Transfer in Goal Learning from Demonstration." In Proceedings of the International Conference on Development and Learning (ICDL), 2011.
8. Chang, Y., T. Levinboim, V. Rajan, & R. Maheswaran. (2011). Learning and Evaluating Human-Like NPC Behaviors in Dynamic Games. In Artificial Intelligence and Interactive Digital Entertainment.
9. S. Chernova and M. Veloso. Interactive Policy Learning through Confidence-Based Autonomy. Journal of Artificial Intelligence Research, 34, 2009.
10. Crick, S. Osentoski, G. Jay, and O. Jenkins, "Human and robot perception in large-scale learning from demonstration," in ACM/IEEE International Conference on Human-Robot Interaction (HRI 2011), 2011.
11. Merrill, Miles. Where's Massive?: Past, Present and Future for The Lord of the Rings' Crowd Software. Metro Magazine: Media & Education Magazine, No. 139, 2004.
12. Nareyek, P. (2007). Game AI is dead. Long live game AI! IEEE Intelligent Systems, 22(1):9-11.
13. Permar, J & Magerko, B. (2013) A conceptual blending approach to the generation of cognitive scripts for interactive narrative. Ninth Artificial Intelligence and Interactive Digital Entertainment Conference.
14. Pew, R. W., & Mavor, A. S. (Eds.). (1998). *Modeling human and organizational behavior: Applications to military simulations*. Washington, DC: National Academy Press.
15. S. Osentoski, B. Pitzer, C. Crick, G. Jay, S. Dong, D. Grollman, H. B. Suay, and O. C. Jenkins, "Remote robotic laboratories for learning from demonstration," International Journal of Social Robotics, vol. 4, pp. 1-13, 2012.
16. M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *Proc. Open-Source Software workshop of the International Conference on Robotics and Automation (ICRA)*, 2009.
17. Rodgers, S., Myers, C., Ball, J. & Freiman, M. (2012). Toward a situation model in a cognitive architecture. *Computational and Mathematical Organization Theory*. (DOI: 10.1007/s10588-012-9134-x)

18. Toris, R., Kent, D., Chernova, S. (2014) *The Robot Management System: A Framework for Conducting Human-Robot Interaction Studies Through Crowdsourcing*. Journal of Human-Robot Interaction.
19. M. Vondrak, L. Sigal, J. Hodgins, and O. Jenkins, "Video-based 3D Motion Capture through Biped Control," ACM Transaction of Graphics (TOG) (Proceedings of ACM SIGGRAPH), vol. 31, iss. 4, 2012.
20. Wallace E. Lawson, J. Gregory Trafton: Unposed Object Recognition using an Active Approach. VISAPP (1) 2013: 309-314
21. Zacharias, G. L., MacMillan, J., & Van Hemel, S. (Eds.). (2008). *Behavioral modeling and simulation: From individuals to societies*. Washington, DC: National Academy Press.