Appears in: Working Notes of the AAAI Workshop on Agent Modeling
Portland, Oregon, August 1996, pp. 112-118.

Agent and Task Modeling at Honeywell

David J. Musliner & Christopher A. Miller
Honeywell Technology Center
3660 Technology Drive
Minneapolis, MN 55418
{musliner,cmiller}@src.honeywell.com

Introduction

Modeling the activities, knowledge, and requirements of agents (both human and automated) has been recognized as a recurrent theme in research conducted by members of the Software and Systems Technology Area (SSTA) at the Honeywell Technology Center (HTC). In fact, we have formed a special interest group that aims to collate and clarify the broad range of task and agent modeling experiences of the hundred-odd SSTA members. This paper gives a high-level overview of the various functions agent modeling has played in our research, as exemplified in several projects. We hope that this effort will eventually lead to fruitful cooperative activities with other researchers pursuing related work.

Why Honeywell Cares

HTC is Honeywell's main corporate research center, tasked with researching cutting-edge technologies related to the main corporate lines of business, including industrial control systems, space and aviation control products, and home and building control systems. Many of the advanced concepts in control systems we investigate have one or more of the following critical aspects that make agent/task modeling essential:

Distribution, making it imperative that different distributed control centers have useful models of what can be done by others, and what they are planning to do.

High-Level Automation, requiring powerful techniques for specifying automated behavior.

Human Interaction, making it essential for the system to understand what the human can do, is doing, should do, needs to know, can assimilate, etc.

Defining the Problem(s)

The general term "agent modeling" subsumes a large number of modeling tasks including representing and manipulating:

- Capabilities (what an agent can do).
- Desires/goals (what an agent wants to do).
- State information (what an agent knows and is doing).

At HTC, the applications-oriented nature of our work has tended to focus projects on these problems with a particular slant that we loosely label "task modeling." As the name implies, the focus tends to be on modeling the tasks an agent can perform, as opposed to the agent-centered focus implied by "agent modeling." Still, many of the modeling issues are similar or identical. Task modeling is generally aimed at describing and reasoning about what tasks need to be done, what resources they take, who is in charge of them, how automation can do them, what human operators need to know in order to do them, etc. In other words, it's a very practical orientation centered around accomplishing some tasks in a given domain, as opposed to being centered around enabling capabilities in agents. Task modeling is also used for widely varying purposes ranging from static analysis of a system design to automating the system design process to actually driving the behavior of a dynamic control system. Perhaps the best way to describe what we mean by task modeling is through examples; the following sections of this paper describe some of the major classes of task information we model, the tools we use, and examples of the related research projects.

Modeling for a Single Human

Some of the earliest uses of task modeling focus on modeling the behaviors of a single human engaged in various tasks. Such task models can be used during the design process for complex human interfaces and other systems, to facilitate effective human-centered design. Many fairly generic software tools are available to capture models of human tasks and analyze those models to extract various performance features with respect to operational environments and candidate system designs. For example, the W/Index and CREWCUT tools can be used to study the expected performance qualities of an aircraft cockpit design (Duley *et al.* 1994). Designers input descriptions of the tasks the human can perform, the interface capabilities the cockpit design provides, and a "scenario" that drives the tool's analysis, simulating the human engaging in a series of tasks. The tools output workload analyses showing how busy the various human capabilities (e.g., visual perception, cognition) were at various times during the scenario (see Figure 1).

The Lab Notebook project, part of Honeywell's

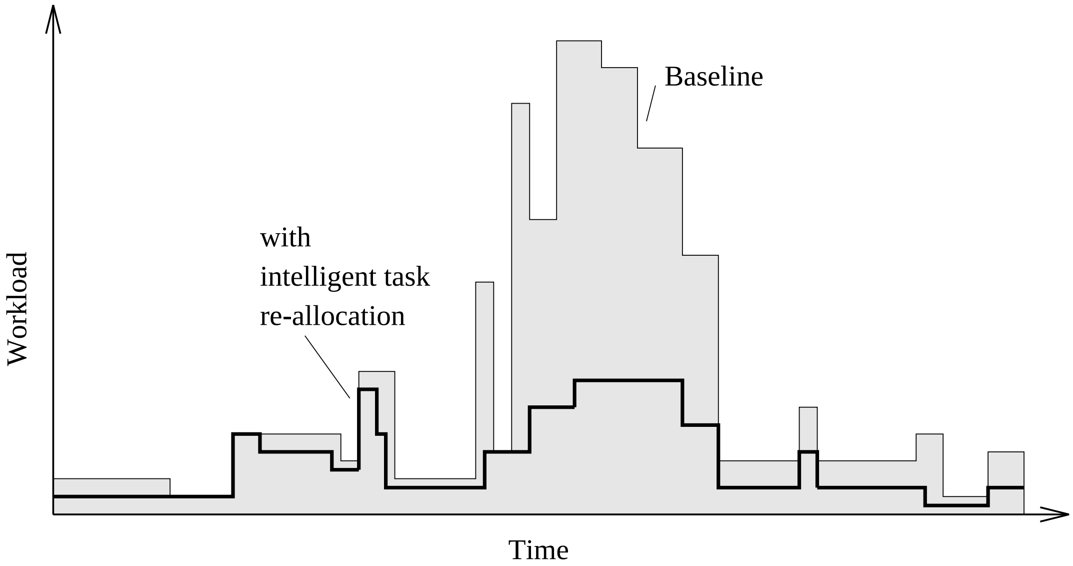

Figure 1: Example RPA workload profile with and without intelligent task reallocation between pilots.

Prototyping Technologies (ProtoTech) program, also exemplifies the use of task modeling information in a static, design-time role. Prototech is an ARPA-sponsored effort to develop a next-generation environment for the rapid assembly of large software prototypes. Our Prototech environment includes a Laboratory Notebook tool that helps engineers record their design decisions and rationales. The Lab Notebook tool is based on a model of human question asking and answering and is designed to maximize communication between members of a design team while minimizing engineer workload in recording rationales. This is accomplished via an interview process which provides a simple, familiar structure on natural language rationales. This structure can be used for later retrieval and navigation of the recorded rationales.

Task models of a single human can also be used in a much more dynamic, run-time environment to control the behavior of an interactive system. For example, the Pilot's Associate (PA) system, sponsored by DARPA and the U.S. Air Force, contained models of many tasks that pilots were expected to perform (Miller & Riley 1994). During a flight, the PA actively monitored the human pilot's activities and attempted to match them against its task models, in order to understand what information needs the human had, and what tasks the PA should perform autonomously when, for example, the human was too busy. One successor to the PA, the Rotorcraft Pilot's Associate, is described in more detail in Section .

Modeling for Multiple Humans

Significant extensions to a single-agent modeling method are required when capturing a description of multiple agents interacting or cooperating. For example, HTC has a NASA Advanced Air Transport Technologies (AATT) Topical Area Study program titled "Analyzing the Dynamics of a Next-Generation Air Transportation Management System". This one-year program is investigating methods for improving human performance simulation tools at NASA and Honeywell (such as NASA's MIDAS and HTC's W/Index) for evaluating candidate advanced air transportation technology concepts long before working prototypes are available for human-in-the-loop studies. There are two thrusts within this program. First, we are developing techniques, utilizing HTC's Mixed-Initiative Model of human/machine interactions (Riley 1989), for extending current simulation tools so that they can consider the information flow between multiple distributed intelligent actors— enabling them to better analyze the complex interactions between ground crews, multiple air crews, and intelligent automation systems which will characterize most advanced air traffic management concepts. The Mixed-Initiative Model, illustrated in Figure 2, breaks out the various perceptual, cognitive, and actuation capabilities of each agent, expressing functional behaviors and interactions between these capabilities. In essence, these capabilities represent modular sub-elements of a single agent's behavioral model, and we are starting to use task modeling methods to specify each capability.

Second, we are working to make the use of human/machine performance simulations more affordable by beginning the construction of a knowledge-based suite of model-building and analysis tools around the simulation systems themselves. In this program we are concentrating on the construction of an object-oriented, graphical model-building tool to facilitate the construction of simulation scenarios and expected agent tasks/behaviors. This tool will be built using HTC's Domain Modeling Environment (DoME) and will instantiate an "information model" of the concepts and entities used to construct mixed-initiative system

Figure not available online

Figure 2: The Mixed-Initiative Model of human-machine systems [from (Riley 1989)].

performance simulations. One piece of this information model will be a set of task modeling structures and forms suited to expressing the behavior of each MIM capability.

Modeling for Automation

Much of the AI community's work on "agent modeling" is focused on capturing the capabilities and behaviors of fully automated systems. At HTC, many of the real-world programs are aimed at developing mixed-initiative systems, and thus modeling for automation alone is less common. However, several technology areas have consistently involved models for fully automated systems. For example, our work on constraint-based scheduling systems uses complex task models to describe the control semantics and temporal details of a set of tasks to be scheduled on a set of resources (e.g., petrochemical processing units). The SAFEbus™ scheduler, for example, builds static processing schedules for a networked multiprocessor system that controls the Boeing 777 aircraft information management system. The scheduler is given a detailed model of the set of tasks to be executed, their periods, jitter, latency, and other temporal constraints, and various additional information such as precedence requirements. Using this task model, the scheduler uses an iterative constraint envelope scheduling technique (Boddy, Carciofini, & Hadden 1992) to develop a schedule that meets all the requirements. Although this program initially seems less related to agent modeling, the complex task modeling capabilities (particularly the expressive temporal constraint language) are closely tied to traditional AI planners and plan representations.

HTC also has projects that use models of autonomous behavior in the prescriptive plan-and-execute fashion common in AI. For example, one project is pursuing an agent-based approach to human-computer interaction and dynamic generation of context-sensitive displays. In this program, the UM-PRS (Lee *et al.* 1994) implementation of the Procedural Reasoning System (Georgeff & Ingrand 1989; Ingrand, Georgeff, & Rao 1992) is being used to capture and reactively execute the procedures involved in a search-and-rescue (SAR) domain. UM-PRS provides an underlying task representation and syntax, and complex task models are instantiated to express the numerous SAR activities. In addition, the task models describe how to generate effective displays based on the current context and available display resources. Thus explicit task modeling serves to capture several levels of the system's behavior.

Modeling for Mixed-Initiative Teams

As noted above, many HTC projects involve complex mixed-initiative systems in which humans and automation systems share responsibility and control. Task models play a variety of roles in these systems, ranging

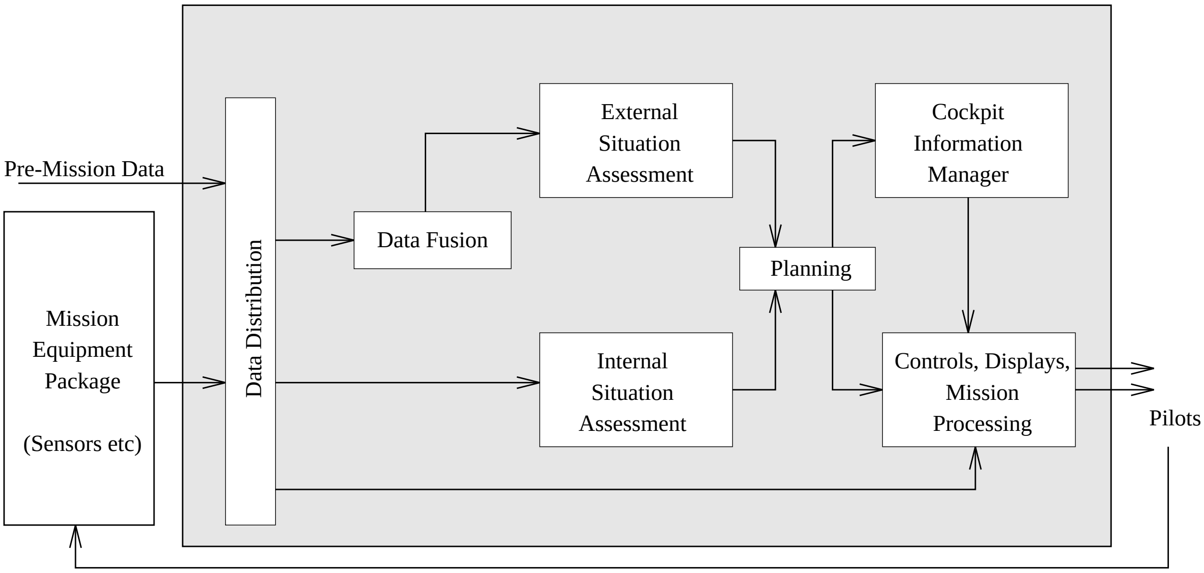
Pre-Mission Data
Mission Equipment Package (Sensors etc)
Data Distribution
Data Fusion
External Situation Assessment
Internal Situation Assessment
Planning
Cockpit Information Manager
Controls, Displays, Mission Processing
Pilots

Figure 3: The RPA Cognitive Decision Aiding System Architecture.

from the design-time analysis illustrated in the above AATT example to specification of run-time behavior to the target information form for learning systems.

Rotorcraft Pilot's Associate (RPA)

The RPA program is a five-year Advanced Technology Demonstration (ATD) sponsored by U.S. Army's Aviation Applied Technology Directorate to develop, extensively evaluate, and flight-test an intelligent associate system in the cockpit of an advanced Army attack helicopter. The objective of this program is to establish revolutionary improvements in combat helicopter mission effectiveness through the application of knowledge-based systems for cognitive-decision aiding and the integration of advanced pilotage, target acquisition, armament and fire control, communications, controls and displays, navigation, survivability and flight control equipment. Similar to the Pilot's Associate programs, the RPA will consist of a suite of five cooperating knowledge-based systems whose collective goal will be to assist the aircraft crew in understanding the vast array of battlefield information, planning the mission, and managing the complex systems of an advanced helicopter (McBryan & Hall 1994). Figure 3 illustrates the overall architecture for the RPA system. Task modeling plays significant roles in several of the system modules and their inter-module communications.

Honeywell is participating in the design, development and evaluation of the Cockpit Information Manager (CIM) module (Miller & Riley 1994). Overall, the RPA architecture relies on task modeling to coordinate its cooperating knowledge-based systems. The situation assessment modules must recognize and announce the tasks that the pilots are pursuing, as well as the external situation, potentially including the task plans being pursued by other entities in the domain. The planning module uses this information to generate a projected course of action for the automation system and the humans, and again this is essentially expressed as a task model. Then the CIM uses this task modeling information to actively manage the cockpit displays and interaction mechanisms. Figure 4 illustrates the CIM architecture and shows several roles for task models, including the inputs to the CIM system, the outputs of the crew intent estimator module, and the stored elements of the goals/side-effects modeling module.

At HTC, we are using a set of workload assessment and human factors analysis tools to optimize crewstation design and function allocation policies between multiple crew members and automation during the initial phases of the program. In addition, the policies developed using these analysis tools will be captured in dynamic information management and function allocation algorithms to increase the flexibility and context-sensitivity of the CIM.

Learning Systems for Pilot Aiding (LSPA)

The LSPA program was a 3.5 year effort sponsored by AFWAL's Crewstation Directorate to demonstrate machine learning applications for large scale, pilot-aiding expert systems— specifically, Lockheed's Pilot's Associate. The program consisted of two interacting parts, Learning System for Tactical Planning (LSTP) and Learning System for Information Requirements (LSIR). Our goals were to facilitate knowledge acquisition and knowledge engineering by semi-automatically learning tactical plans and pilot information requirements from simulator-flown learning instances, thereby reducing the time and cost associated with knowledge base scale-up and modification. At program end, LSTP has successfully used an Explanation-Based Learning approach to learn eight new leaf-level plans for the

Figure not available online

Figure 4: The RPA Cockpit Information Manager.

Tactics Planner (TP) module of the PA, substantially fleshing out the TP branches for degrading SAM sites and evading missiles. LSIR has developed a "linked learning" technique which takes a newly-learned plan from LSTP and reasons about the information a human pilot will need in order to perform that plan. The result is an Information Requirements data structure which is used by the Pilot-Vehicle interface module of the PA in selecting and configuring displays for Information Management. Experimental evaluation of the LSPA systems indicates that when human knowledge engineers use LSPA outputs, even novices can create plans and IR data structures faster and with greater accuracy and completeness than when relying on traditional knowledge engineering techniques alone.

Explanations for Model-Based Systems

The purpose of this IR&D project was to provide explanation and "argumentative" capabilities for a model-based diagnostic expert system. The target application was Honeywell's model-based Flight Control Maintenance and Diagnostic System. We developed an approach to organizing the presentation of large amounts of model-based data in an interactive format patterned after human-human explanatory and argumentative discourse in order to increase user trust, accuracy of usage, and embedded training potential. The discourse approach was a convenient, powerful, intuitive and broadly applicable method of organizing model-intensive data for information exchange in human/machine and human/human interactions.

Summary

These examples display the broad scope of task modeling and agent modeling efforts at the Honeywell Technology Center. Many of these projects have a more human-centered focus than current AI software agent modeling work, but we see many common themes and approaches. In the long term, we hope to develop shared agent modeling representations and tools that can fulfill several of the roles currently addressed by separate techniques. At HTC, task modeling supports many roles including:

- Automated control and associate systems, through dynamic planning, scheduling, and execution of task models.
- Domain knowledge acquisition, through models of discourse and communication, as well as abstract task models representing structural constraints on the target knowledge being acquired.
- Intelligent tutoring systems, using task models as both the object to be trained and as a guide for how to provide training.
- Intent inferencing (plan recognition), by mapping observed user activities against existing task models.
- User interface generation, using task models as the basis of determining what information a user needs.

- Interface design and analysis, using explicit task models for domain-specific scenarios as the testbed against which to evaluate various hypothetical interface designs.
- Information management and interaction management systems, which may build task models dynamically (in conjunction with an intent inferencing or task tracking system), to provide adaptive assistance to human operators.

References

Boddy, M.; Carciofini, J.; and Hadden, G. 1992. Scheduling with partial orders and a causal model. In *Proceedings of the Space Applications and Research Workshop*.

Duley, J. A.; Miller, C. A.; Schultz, E.; and Hannen, M. 1994. A human resource based crew station design methodology. In *Proc. AIAA/IEEE Digital Avionics Systems Conference*, 233–238.

Georgeff, M. P., and Ingrand, F. F. 1989. Decision-making in an embedded reasoning system. In *Proc. Int'l Joint Conf. on Artificial Intelligence*, 972–978.

Ingrand, F. F.; Georgeff, M. P.; and Rao, A. S. 1992. An architecture for real-time reasoning and system control. *IEEE Expert* 34–44.

Lee, J.; Huber, M. J.; Durfee, E. H.; and Kenny, P. G. 1994. UM-PRS: An implementation of the procedural reasoning system for multirobot applications. In *Proc. AIAA/NASA Conf. on Intelligent Robots in Field, Factory, Service, and Space.*

McBryan, B., and Hall, J. 1994. Engineering approach for rotorcraft pilot's associate cognitive decision aiding system development. In *Proc. AIAA/IEEE Digital Avionics Systems Conference*, 76–81.

Miller, C. A., and Riley, V. 1994. Achieving the associate relationship: Lessons learned from 10 years of research and design. In *Proc. IEEE Int'l Conf. on Systems, Man and Cybernetics.*

Riley, V. 1989. A general model of mixed-initiative human-machine systems. In *Proc. Human Factors Society 33rd Annual Meeting*, 124–128.