Multi-Agent Activity Modeling with the Brahms Environment

Maarten Sierhuis (sierhuis@parc.com)
Palo Alto Research Center, 3333 Coyote Hill Road
Palo Alto, CA 94304 USA

Keywords: agent-based modeling, BDI, cognitive modeling,

Introduction

More and more people are interested in developing "day in the life" models and simulations of people's behavior at the second and longer timeframe, the interaction between groups of people and systems, as well as the movement and interaction within the environment. Cognitive modeling tools (e.g. SOAR, ACT-R) focus on detailed modeling of individual cognitive tasks at the sub-second level. In contrast, activity modeling focuses on higher-abstraction behaviors that enable modeling of people's daily activities and enable a focus on how informal, circumstantial, and located behaviors of a group of individuals occur and where communication and synchronization happen, such that the task contributions of people and machines flow together to accomplish goals. This is referred to as "work practice modeling."

Brahms includes an activity-oriented Belief-Desire-Intention (BDI) language, a compiler and virtual machine for executing Brahms models, as well as an Eclipse plug-in and a post-execution viewer of agent execution, communication and interaction. Brahms enables the creation of multi-agent models that include aspects of reasoning found in cognitive models, task execution, plus the impact of interaction and geography, such as agent movement and physical changes in the environment. Brahms is currently used to automate the work of a flight controller in NASA's International Space Station's Mission Control Center (ISS MCC). This system, called OCAMS, has been in production in the ISS MCC, 24x7, since July of 2008, and is based on a Brahms model of the work practices of the flight controllers. OCAMS is a distributed Multi-Agent System (Sierhuis et al., 2009b).

Motivation for Brahms

Brahms was developed as a multiagent modeling and simulation language to visualize the social systems of work for business redesign projects. The Brahms language was originally conceived of as a language for modeling contextual behavior of groups of people, called work practice.

> *Work Practice:* The collective performance of contextually situated activities of a group of people who coordinate, cooperate and collaborate while performing these activities synchronously or asynchronously, making use of knowledge previously gained through experiences in performing similar activities.

This created two very important ideas for the language: First, to model a group of people it is very natural to model them as software agents. Second, modeling situated behavior of a group imposes a constraint on the level of detail that is useful in modeling the dependent and independent behavior of the individuals. The right level is a representational level that falls between functional process models and individual cognitive models (Clancey et al., 1998). If we are interested in modeling a day-in-the-life of say ten or more people, modeling the individual behavior at the level of cognitive task models will be very time consuming, because these models are generally at the millisecond decision-making level. To overcome this kind of detail, the Brahms language uses a more abstract level of behavioral modeling that is derived from Activity Theory (Leont'ev, 1978; Vygotsky, 1978) and Situated Action (Suchman, 1987). An individual's behavior is represented in terms of activities that take an amount of discrete time and can be decomposed into more detailed subactivities if necessary.

Brahms demonstrates how a multiagent belief-desire-intention (BDI) language, symbolic cognitive modeling, traditional business process modeling, activity- and situated cognition theories are brought together in a coherent approach for analysis and design of organizations and human-centered systems.

The Brahms Language

The Brahms language is a pure AOL (Sierhuis et al., 2009a). It is not a set of Java libraries enabling agent-based programming in the Java language. Instead, Brahms is a full-fledged multiagent language allowing the modeler to easily and naturally represent *multiple agents*. Although Brahms was originally developed for modeling people's behavior, the Brahms language is a domain independent language. This means that the modeler decides what a Brahms model represents. Agents can represent whatever autonomous entity the modeler wants to represent, such as a person, an animal, or an autonomous or intelligent system. Brahms includes the following language features:

- *Mental attributes*: attributes, relations, beliefs and facts.
- *Deliberation and Intention*: concluding new beliefs, and use of production rules for reasoning.
- *Adaptation*: changing beliefs, execution activity behavior and reasoning based on context.
- *Social Abilities*: groups and group inheritance, communication, and modeling the environment (objects, geography and location).

- *Reactive and Cognitive-based behavior*: modeling activity behavior, versus pure cognitive behavior, detectables, workframe-activity subsumption.
- *Agent Communication*: communication activities, and communicative acts.

Brahms is an agent-oriented BDI-like language. It allows easy creation of groups of agents that execute parallel activities based on local beliefs. Below is a simple taxonomy of the language concepts:

GROUPS are composed of
 AGENTS having
  BELIEFS and doing
  ACTIVITIES executed by
   WORKFRAMES defined by
    PRECONDITIONS, matching agents beliefs
    PRIMITIVE ACTIVITIES
    COMPOSITE ACTIVITIES, decomposing the activity
    DETECTABLES, including INTERUPTS, IMPASSES
    CONSEQUENCES, creating new beliefs and/or facts
  DELIBERATION implemented with
   THOUGHTFRAMES defined by
    PRECONDITIONS, matching agents beliefs
    CONSEQUENCES, creating new beliefs

Cognitive Modeling in Brahms

Brahms borrows some of the theoretical underpinnings of the ACT-R theory about human knowledge. Just as in ACT-R, Brahms assumes there are two types of knowledge—declarative and procedural.

Declarative knowledge in Brahms is represented as "beliefs" of *individual agents*. A *belief* of an agent in Brahms plays a similar role in processing the procedural knowledge of the agent as chunks do in ACT-R, i.e. they are matched to preconditions of rules. However, beliefs are semantically and syntactically simpler than chunks. A belief is a first-order predicate statement.

Brahms represents the *procedural knowledge* of an *individual agent* as rule-like constructs called *workframes* and *thoughtframes*. The condition-part—called *pre-conditions*—are matched against the belief-set of the Brahms agent. When all the preconditions of a workframe or thoughtframe match, the rule is put onto the agent's work stack. Each of these rule-types is processed independently by the virtual machine. Hence the reason for separate stacks.

Thoughtframes are similar to production rules in ACT-R, in that their action-part consists *only* of changes to, or additions of beliefs in the agent's declarative memory, i.e. its belief-set. Thoughframes are executed in a forward-chaining mode. Workframes are the main type of rule in Brahms. Activities are executed in the action-part of a workframe (the workframe's *body*). The time it takes to fire a workframe depends on the total time it takes to execute all of the activities in its body. The workframe rule-type in Brahms corresponds to the goal-directed production rules in ACT-R, with the addition that in a workframe we can

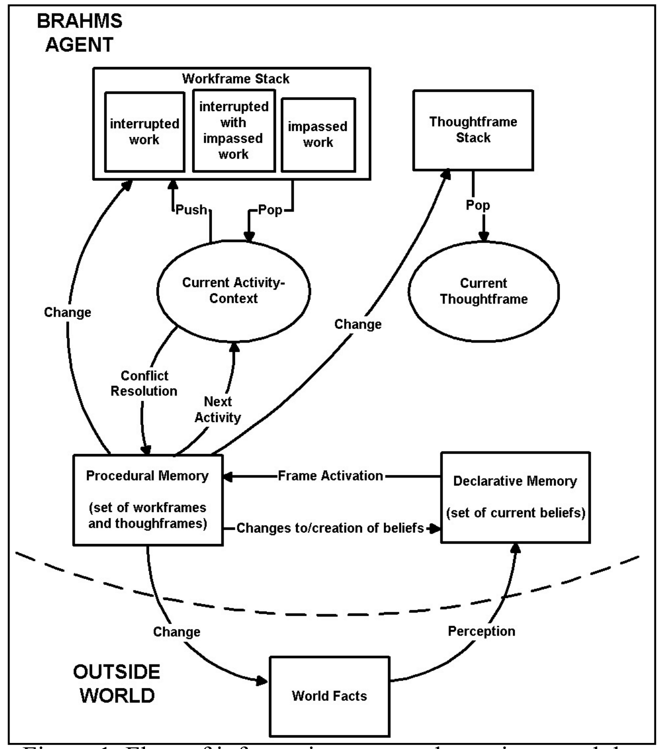

Figure 1. Flow of information among the various modules in a Brahms Agent

include the changes in the external world, and not only the internal declarative changes in the memory of the agents.

We show a similar figure (Figure 1) for a Brahms agent as the figure in Anderson's book about ACT-R (Anderson and Lebiere, 1998)

References

Anderson, J. R. and Lebiere, C. (1998) The atomic components of thought. Mahwah, NJ.: Lawrence Erlbaum Associates.

Clancey, W. J., Sachs, P., et al. (1998) "Brahms: Simulating practice for work systems design", International Journal on Human-Computer Studies 49: 831-865.

Leont'ev, A. N. (1978) Activity, Consciousness and Personality. Englewood Cliffs, NJ: Prentice-Hall.

Sierhuis, M., Clancey, W. J. et al. (2009a) "Brahms: An Agent-Oriented Language for Work Practice Simulation and Multi-Agent Systems Development ", in M. D. Rafael H. Bordini, Jürgen Dix, Amal El Fallah-Seghrouchni (ed.) Multi-Agent Programming, 2nd Edition: Springer.

Sierhuis, M., Clancey, W. J., et al (2009b) "NASA's OCA Mirroring System: An application of multiagent systems in Mission Control", 8th Int. Conf. on Autonomous Agents and Multiagent Systems 2009. Budapest, Hungary

Suchman, L. A. (1987) Plans and Situated Action: The Problem of Human Machine Communication. Cambridge, MA: Cambridge University Press.

Vygotsky, L. S. (1978) Mind in Society: The Development of Higher Psychological Processes. Cambridge, MA: Harvard University Press.