Cognitive Science 25 (2000) 611–657

COGNITIVE SCIENCE

http://www.elsevier.com/locate/cogsci

Convention in joint activity

Richard Alterman*, Andrew Garland

Computer Science Department, Volen Center for Complex Systems, Brandeis University, Waltham, M.A. 02254, USA

Abstract

Conventional behaviors develop from practice for regularly occurring problems of coordination within a community of actors. Reusing and extending conventional methods for coordinating behavior is the task of *everyday reasoning.*

The computational model presented in the paper details the emergence of convention in circumstances where there is no ruling body of knowledge developed by prior generations of actors within the community to guide behavior. The framework we assume combines social theories of cognition with human information processing models that have been developed within Cognitive Science. The model presented reflects both elements of the framework. Conventional behaviors are partially coded in the predisposition of participants in a joint activity to expect certain *points of coordination* to develop during the course of the activity. The expected points of coordination that are commonly assumed form a *design for an activity.* Because of uncertainty, interruptions, and numerous other opportunities to get off-track and out-of-synch, the participants must work jointly and continuously to achieve conventional coordination.

One feature of the model is that the community improves its performance despite the fact that individual actors reason independently about their experiences. Another important feature of the model is that the mechanisms for improving behavior are tied to the memory function of individual actors. A third important feature is that the social interaction among the participants simplifies and drives the everyday reasoning processes. An analysis of a large set of computational experiments supports the theoretical position that is developed regarding everyday reasoning and convention.

* Corresponding author. Tel.: +1-781-736-2703.

E-mail address: alterman@cs.brandeis.edu (R. Alterman).

PII: S0364-0213(01)00047-7

1. Introduction

Most everyday behavior is done jointly with another actor. Performance depends on cooperation and coordination among the participating actors. Conventional behaviors develop for recurrent problems of coordination within a community of actors, reducing the effort and work required to coordinate.

One example of a coordinated behavior that has its basis in convention is crossing a road at an intersection either by car, by bicycle, or by foot. Participants in this situation are informed by a large set of conventions concerning diverse things such as crosswalk markings, multiple cars approaching a stop sign, ordering preferences between turning cars, differing configurations of merging traffic (T-intersections and Y-merging thoroughfares) and types of roads (highways and residential streets, or one-way and two-way roads), ages and capabilities of pedestrians and drivers, et cetera. Deciding which of these policies are in force requires coordination between individuals, and conventions also exist for making that decision. Any sort of stereotypical situation for which a script could be generated to represent the coordination of behavior between participants is also a case where convention is at work, for example, subway stations, restaurants, buses, and the supermarket checkout. Using devices, tools, and artifacts is primarily reasoning about convention. Communication, in any of its forms, is tied to conventions of use.

Reusing and extending conventional methods for coordinating behavior is the task of *everyday reasoning* and the subject of this paper. The framework we assume combines social theories of cognition with human information processing models that have been developed within Cognitive Science. Within Cognitive Science, this framework for studying cognition has been explicitly argued for by Hutchins (1995a b) and Greeno (1998). The foundations of the social theories we assume include the works of Mead (1934) on cognition as a social act, the Soviet cultural and historic models of cognition (Vygotsky, 1978; 1986) and their American counterparts (Cole, 1996), and recent work that assumes activity and interaction (Suchman, 1987; Lave, 1988; Agre, 1997) —instead of cognitive structure —as the basic unit of analysis in the study of cognition. The tradition of research on human information processing we assume originates with the work on human problem solving of Newell & Simon (1972). Within that tradition, Artificial Intelligence models of planning that began with the work on STRIPS (Fikes & Nilsson, 1971 and models of reasoning that are tied to memory or cases (Schank, 1982; Kolodner, 1993) are most directly relevant, especially the work on adaptive planning (Alterman, 1988) and everyday reasoning as pragmatic action (Alterman et al., 1998; Alterman, 1999). Also of direct relevance is cognitive work on the design of everyday things (Norman, 1988).

The model we present reflects both elements of the unified cognitive science and social theory framework. From the perspective of individual psychology:

- Reasoning about everyday joint behavior is a memory-based process tied to pragmatic action. The predisposition for certain points of coordination to develop during the course of activity, and the assumption that other actors will share those predispositions, frame the individual's behavior.

From the perspective of a social theory of cognition

- The social interaction among participants circumscribes and simplifies the reasoning processes. The participants agree to points of coordination as they proceed. The historic elements of the social interaction between actors within a community cannot be factored out of the analysis of conventional behavior.

Crossing the street at a busy intersection exemplifies each of these features of the model. The reasoning of pedestrians, cyclists, and drivers alike depends on the ongoing social interaction within the community. Eye contact, and other forms of communication, reduce a participant's effort in interpreting the actions of other actors. Reasoning in the situation depends on familiarity with the behaviors, types of drivers (e.g., taxi drivers or school bus drivers), and how they reason about crosswalks and yellow lights (artifacts) in a given community. Some of the structure of the joint behavior is coded in expectations of the actors, but the actual structure is realized only during the social interaction.

1.1. Overview of the paper

The first part of the paper frames the discussion of convention. Matters concerning joint cognition and behavior are discussed in the terms of Clark's (1996) model of *joint activity.* Lewis' (1969) definition of convention is presented. Scripts (Schank & Abelson, 1977) is discussed as an example of a cognitive theory about conventional behavior. Research on situated activity (Suchman, 1987) and distributed cognition (Hutchins, 1995a) is used to critique the script model of reasoning about conventions of behavior and is also used to develop a more interactive and situative model of everyday reasoning about conventional behavior. Points of coordination, designs (and structures) for activity, and the distinction between expectations about joint behavior and realizations of joint behavior are some of the critical features of the analysis.

The second part of the paper explores the emergence of convention in circumstances where there is not a ruling body of knowledge developed by prior generations of actors within the community to draw on as potential designs for cooperative and coordinated behaviors. Our example domain (MOVERS-WORLD) is a group of actors who are part of a moving company. Their job is to move boxes and furniture from a house into a truck. Actors start off with very little knowledge of each others capabilities, behavioral patterns, or roles. With practice, individuals within the community begin to converge on a set of conventions for behaviors that match the regularly occurring problems of coordination in the domain of activity. The community improves its performance despite the fact that individual actors reason independently about their experiences. Because the actors use prior experiences to guide future behaviors, they can, however, indirectly adjust their expectations about the best way to coordinate behavior by making adjustments that reflect the success and failure of this or that act. Memory for joint activity is organized around *points of coordination* that occurred during prior activities. The middle of the paper works through some of the details of our architectural assumptions.

The third part of the paper presents a set of experiments that explore some features of convention within the context of MOVERS-WORLD. The first set of experiments examines the performance changes that occur as conventions of behavior develop to match the pragmatics of the domain. Conventions reduce the amount of work (number of actions) to achieve a set of common goals. Conventions reduce the amount of planning and communication that individuals need to do in order to stay coordinated. The second set of experiments analyzes how memory for prior coordinated activity contributes to improvements in performance. These latter experiments show that MOVERS-WORLD actors are learning to orient themselves to a situation in compatible manners, that is, they are working on related goals and develop plans with similar expectations about points of coordination. One hypothesis —that an ideal script for a conventional behavior is being determined in the individual minds of participating actors —is rejected.

The paper concludes with a lengthy discussion section. After returning to the theme of a unified framework for the study of cognition, three questions are considered: Can the entire structure of conventional behaviors be reduced to a unique idealized form accessible independently by each participant prior to activity? Are the processes that compose everyday reasoning identical to those of analytic and scientific reasoning? Are two minds reducible to one?

2. Joint activity and convention

2.1. Joint activity

Joint activities are essentially a complex of coordination problems (Clark, 1996). Playing a duet, shaking hands, rowing a boat in tandem, eating dinner at a restaurant, talking with a friend, and so on, are all joint activities that require that the participants solve a set of coordination problems.

Joint activities have participants, who assume public roles. There are joint public goals and private ones. Joint activities emerge as a hierarchy of joint actions and joint (sub)activities. The entry and exit boundaries of a joint activity are jointly engineered by the participants. Joint activities advance one increment at a time, mostly through *joint actions*. Joint actions are created when people coordinate with each other. Joint actions have phases that have entry and exit points, each of which require coordination. Coordination on the entry and exit times to each phase can be achieved by means of different strategies.

An example of a joint activity is Sam and Gladys, two teenagers on a date engineering a 'first kiss'. Sam and Gladys are the participants. Sam's public role is 'Romeo' and Gladys' public role is 'Juliet'. Their joint public goal is to kiss; their private goals may differ. Entry into the first stage of kissing is jointly engineered by the participants. Sam and Gladys are doing their parts in the joint action of initiating a kiss. Coordination of entry and exit times for each of the phases, from the initial touch, to looking into each others eyes, to the first touch of lips, is jointly achieved. The joint action (activity) of Sam and Gladys kissing for the first time is a complex of coordination problems.

What accumulates during a joint activity is common ground. At any moment during a joint activity, common ground has three parts: initial common ground, current state of the joint activity, and public events so far. The initial common ground is the set of background facts, assumptions, and beliefs presupposed at the outset of the joint of activity. The current state of the joint activity is where in the activity the participants presupposed themselves to be. The public events are those events presupposed by the participants as leading up to the current state of affairs.

The initial common ground for Sam and Gladys' kissing includes the cultural history within their community for the activity of 'first kissing' and the prior dating behaviors of their clique. As Sam and Gladys proceed through the activity, changes in common ground slowly, inexorably, move from assumptions and beliefs that the 'first kiss' might happen to the confirmation that it has. Events like Sam putting his hand on Gladys' shoulder, and Gladys reciprocating by putting her hand on his hip, are 'public' events which mark the progress of the couple in their joint activity.

2.2. Convention

Lewis (1969) defines convention as a 'solution' to recurrent coordination problem. Conventions are the regularities of behavior that develop among a community of actors with a tradition of common goals and shared activities. Examples of behavioral conventions are:

1. How you answer the telephone: "This is the Alterman residence." Greeting a colleague at work: you say "Hi." Greeting an associate at a meeting: you shake hands. Greeting family you have not seen in awhile: you give them a hug
2. The division of labor between a spouse and spouse: for cleaning up after dinner, she clears the table and puts the leftovers away; he does the dishes. In the morning, she prepares breakfast and he makes bag lunches for the kids
3. Pedestrians crossing in opposite directions at a busy intersection in downtown San Francisco: Say "Excuse me" if you bump into somebody else; tend to walk on right hand side; and stay within the crosswalk
4. Dressing appropriately for a dinner party, for work, for the opera, . . .
5. The procedure you follow for buying lunch at a restaurant
6. Where to meet a student for an appointment. Where to meet your doctor for a physical exam. Where to buy a ticket for a movie, catch the bus, or board an airplane
7. The usage of tools and devices (artifacts): The ways that door knobs are manipulated, the opening and closing of doors, or books, or packages, or containers, taking the cap off a pen, pushing the tab of a ball point pen, opening a CD case, starting up a computer, turning on the light switch, using headphones, calling on a telephone, . . .
8. The manufacturing of any artifact involves conventions about the size, shape, and function of its various parts. For example, there are conventions for sizes, shapes, and strengths of boxes that have developed between the communities of actors who make and use them.

In each of these situations, there is a recurrent coordination problem that is being 'solved'.

Other features of Lewis' definition concern common knowledge and the arbitrariness of convention. Convention is a regularity R in the behavior of members of a given community that occurs in a recurrent situation S (Lewis, 1969; p. 78). Several things must be common knowledge within the community regarding any instance of S. Almost everyone conforms to R and expects almost everyone else to conform to it too. The actors within a given situation

S all have approximately the same preferences for all combinations of actions. Conventions also have an element of arbitrariness. Conventions are arbitrary because there always exists another regularity of behavior for the situation that would work as well. Where a unique solution exists, members of the community conform to it not because it is a convention but because it is the best thing to do.

While we agree with some of the general features of Lewis definition (i.e., common knowledge and the arbitrariness of convention), there are some technical problems with Lewis definition. Most of these problems stem from the lack of an adequate set of cognitive assumptions. Take, for example, the notion of a recurrent situations S. How is it that two actors will agree to what situation S is? In any kind of social situation, it is highly unlikely the participants in the situation will agree a priori to what S is —that it is what is negotiated among the participants as a part of their social interaction. Another problem with Lewis' definition is the notion that a convention is a fixed regularity R. Because of variation in circumstances and continuous change, because of the work that needs to be done to align perspectives among the participants, the idea that the convention has a unique structure R is suspect. The participants can share expectations about the structure of a conventional activity, but the actual structure of the conventional behavior on a given occasion will be uniquely determined on each occasion.

2.3. Internalized, emergent, and/or situative?

Scripts (Schank & Abelson, 1977) is an example of a cognitive theory that is meant to account for convention-based behavior. A script represents a sequence of events coupled to a particular context with actors having roles and coordinating their behavior in predictable ways in order to achieve overlapping goals. Attending a movie, a restaurant, a wedding, or a Bar Mitzvah are all occasions of script-like activity —as are riding a train, flying in a plane, or boarding an oceangoing vessel.

The notion of a script was developed before models of goal-directed behavior were transformed from plan-based to activity-based (Suchman, 1987; Agre & Chapman, 1990). In the mid-80's, models of goal-directed behavior began to account for the uncertainty of the world. Plan-based models assumed a benign world, where things went according to plan: one makes a plan and then executes it. For various reasons, researchers began to consider alternate models of behavior: the intractability of planning (Chapman, 1987), the inadequacies of models of acting that were dependent on internal representations (Brooks, 1991), the binding problem (Agre & Chapman, 1987), and the general uncertainty of the world. Under these newer schemes, plans were not a complete specification of action. Rather, plans were vague, only providing an orientation (Suchman, 1987). Much of the structure in the activity emerged from the individual's interaction with the environment during activity and did not exist prior to the activity (Garfinkel, 1967).

These kinds of critiques also apply to models like scripts. A script is a rigid representation of a particular sequence of events; for everyday behavior, the uncertainty of the world forces a less rigid representational form for knowledge of conventions. Pedestrians crossing in opposite directions at a busy intersection in downtown San Francisco is an example of individuals coordinating their behavior in a regular manner where their joint behavior is not characterizable by a script. Another example is the joint activity of a team of movers. In moving furniture and boxes from the house into a truck, there exist regular patterns of coordination among the crew, but none of these can be reified into a script. In both cases, there is too much uncertainty for events to unfold exactly according to some predetermined script. Even activities that seem more script-like in nature, such as buying lunch at the drive-through at McDonald's, are filled with numerous events that are part of the 'drive-through scenario' but deviate from the specifications of the script. While placing your order, your daughter Emma interrupts to change her order from a Sprite to a Coca Cola, and maybe, on second thought she would prefer a plain cheeseburger to the Chicken McNuggets. Deviations in the main current of the activity are constant and commonplace. A predetermined script that each of the actors follows only accounts for some of the richness of the phenomena.

For a joint activity of buying food at the drive-through at McDonalds, there is more than one expected point of coordination. The term *points of coordination* refers to the points in a joint activity when participants are interacting with one another in order to develop some common course of action. These include requests and responses, which identify and commit actors to joint activities, and waiting, which is used to synchronize the actions between participants. Over time, actors within a given community of practice begin to expect that certain points of coordination will occur during the course of conventional activities. During the technical discussion of memory, points of coordination will be a key feature of our model of individual memory.

Collectively, the points of coordination assumed by the individuals within a community form a *design for the activity*. The ongoing process within a community of actors to create structures for common activities that simplify coordination is the design process. From the perspective of a larger time scale, the structure of conventional behavior is a product of the historical process of design. Designs for conventional activities are reused and extended by means of continued joint activity/social interaction within a community of actors who share a common set of tasks to accomplish.

Internalization of some portions of the design is a prerequisite for generating conventions of behaviors (it is part of the initial common ground), but the design is not a complete specification of what occurs when the convention of behaviors unfold. Whatever expectations are realized require joint effort by the participants. Much of the work they do to keep themselves in-synch and on-track involves helping one another by confirming and realizing each other's expectations through verbal and nonverbal communication. There is no fixed design for a conventional activity. It emerges uniquely on each occasion of activity. Over time the expectations within the community converge somewhat. Nevertheless, because of uncertainty, dynamics, and change, the design never reaches the status of a unique fixed internal structure. This account is both *interactive* (Suchman, 1987; Lave, 1988; Agre, 1997) and *situative* (Hutchins, 1995a; 1995b; Greeno, 1998).

A key feature of an interactive account is that structure emerges from the give-and-take of activity that does not exist prior to the activity. For Suchman (1987; p. 52), a plan for canoeing down a series of whitewater rapids may orient one, but much of the apparent structure of the activity emerges from the activity itself. The work of Sachs et al. (1974) makes the case that turn-taking in conversation is a locally and interactionally managed form of group decision-making that is administered by the participants of the conversation; it emerges from the conversation and is not predeterminable. Lave's work (1988) on the 'dialectic' of a shopper doing arithmetic in the grocery store shows how context is inseparably part of the activity. For Lave et al. (1984; p. 75), one's personal sense of an activity of shopping at the supermarket is not completely internalized; the potential of a given 'setting" is only realized in the dialectic of activity. In Artificial Intelligence, the work of Agre (1997) and Agre & Chapman (1987; 1990) demonstrates how 'plan-like' behavior can emerge without planning from an interaction between the machinery of the individual's reasoning processes and the dynamics of the world. In their PENGI model, the actor continuously redecides what to do; its representations are deictic, thus requiring an interaction to be realized. In each of these cases, cognition and some of the structure of behavior is emerging from activity and interaction.

Conventions of behavior are partially emergent. The participants in a joint activity have a predisposition to act in ways that will simplify coordination. They have familiarity with points of coordination (designs) for certain kinds of activities, but the potential of these designs is realized and emerges only during the activity. The convention of behavior is not uniquely determinable independent of the occasion and procession of a given activity. One time it emerges one way, a second time a different way; certain points of coordination tend to be realized on each occasion, but in different manners. Some of the structure of the activity, of which one would attribute the term 'convention', emerges from the activity itself.

The situative view emphasizes the study of cognition in terms of units of analysis larger than an individual (Hutchins, 1993; p. 62):

> If the individual mind itself is the only locus considered for the structures that organize thinking, then everything that is required to create a sufficient account of cognitive activity has to be crammed into the individual mind. This leads the followers of this view to try to put more in the individual mind than belongs there. The properties of groups of minds in interaction with each other, or the properties of the interaction between individual minds and artifacts in the world, are frequently at the heart of intelligent human performance.

One of Hutchins' example is the airplane pilot and the cockpit. He argues (1995b; p. 286) "The cockpit system remembers its speed, and the memory process emerges from the activity of pilots. The memory of the cockpit, however, is not made primarily of pilot memory." Some significant amount of what needs to be remembered by the airplane pilot is located in representations (e.g., the 'speed bugs' on the airspeed indicator) available in the task environment; thus tasks requiring complex and costly internal memory operations can be distributed. A second example is the navigation of a naval vessel that is underway (Hutchins, 1995a; 1993). There are two parts to navigating a ship: position fixing (which essentially answers the question where are we) and dead reckoning (where are we heading). Position fixing is *distributed* among the ship personnel and their tools. Bearing takers, timer-recorders, plotters, the keeper of the deck log, and the fathometer operator all participate in position fixing. Several actions are coordinated to occur simultaneously at the call of "standby to mark." Much of the computation in fixing the point is done by the nautical chart (the mercator projection chart), which is a specially constructed artifact that supports this computation. The cognition that is occurring is distributed and is part of a larger socio-technical system.

Everyday actors also rely on the larger context to manage the achievement of a conventional behavior. There are no guarantees that two actors' expectations about the course of a joint activity will exactly match. They may have some overlapping expectations about points of coordination, but other parts of the design are contingent on the specifics of the larger context. Novices at a particular kind of conventional activity can rely on the larger context to fill in gaps in knowledge. The first time an individual uses a device, she may realize a convention of behavior, even though she may not be familiar with the type of device she uses. This is because instructions are generally available in the larger context. When the individual dials a number on the air telephone for the first time, information is provided in the design of the setting to aid the user in performing the needed conventions of behavior. The label on one of the buttons indicates 'dial tone', the button affords pushing, other buttons are labeled with numbers. The instructions inform the user she must first push the dial tone button and wait for a dial tone before dialing the number. Other situations, with different arrangements in the design of the interface, require the user to draw on her knowledge of other conventions for operating devices. In each case, the setting of device usage provides signs for the user to interpret in order to determine the situation. Emerging from the development of the activity in the larger context is a convention of behavior that did not pre-exist in the mind of the individual actor, but fits the design for the activity that has been developed within her community of actors.

2.4. The design of an activity

Design is a method for making a device or tool more usable (Norman, 1988). Some features of the design process are (Winograd, 1996; pp. xiii–xxv): The design process creates artifacts that are well-suited to their environments of use. Design activities require the management of trade-offs. "The ongoing process of design is iterative at two levels: iteration by the designer as a piece of current work develops, and iteration by the community as successive generations reveal new possibilities for the medium." Design entails an ongoing dialog between designers and users. Design is a social activity. Users avail themselves of the design as a support for using the artifact; it informs the users' activity (Norman, 1988). Participatory design occurs when the people destined to use the artifact play a critical role in designing it (see Schuler & Namoika, 1993; Greenbaum & Kyng, 1991).

All of these features of the design process for artifacts apply to the design process that coparticipants realize as they produce, with practice, a design for regularly occurring joint activities. The design of a conventional behavior is tied to the emergence of a more reliable set of expectations about the points of coordination most likely to occur in a given situation. The creation of structures for conventional behavior is a participatory design process. Each time coparticipants iterate through a recurring joint activity, improvements in the design can occur. Iteration by successive generations of the community reveal new possibilities for designs of joint activities. Over time, for regularly occurring problems of coordination, a design for the activity develops. A design for a conventional behavior is part of the initial common ground shared within a community of actors. A design may have an external representation (e.g., instructions) but it is mainly coded in the expectations of individuals about expected points of coordination. Knowledge of designs for conventional activities, and access to external representations of it, is a resource that informs the participants' reasoning as they proceed with a joint behavior.

Suppose an actor is helping another actor in a wheelchair pass through a doorway. A spouse Dick recently broke his leg in a skiing accident and he is in a wheelchair; his wife Jane is helping him to get about. From Jane's perspective, a design that coordinates her pushing Dick through a doorway will emerge as the activity proceeds. How to position the wheelchair, when to open the door, and what are reasonable expectations about what participants can and will do, are features of the situation that must be discovered. At the outset, the first time, one does not know a good design for the activity; however, the physical constraints of the situation and constraints that emerge from the actions of each of the participants will shape the joint activity (Norman, 1988); the participants jointly *attune* their behavior to the constraints and affordances of the situation (Greeno, 1998; pp. 8–9). If another couple were in a similar situation, one would expect them to work out a similar design because many of the constraints and affordances that are reasoned about are present in the design and construction of the doors, doorways, and wheelchairs that are used. Through continued practice, Dick and Jane develop a good design for the activity (a convention of behavior). Certain points of coordination become regular features of how they negotiate this kind of joint activity; there are points of coordination in the design that Dick and Jane can reasonably expect will be achieved during the activity.

Whatever design for the activity emerges, it is historically conditioned. Each of the artifacts (i.e., doorways, doors, wheelchairs) has a history of designed activities in which it is used. Whenever actors learn to use a wheelchair, they are part of a continuing story about wheelchair design, construction, and usage. The performance of this activity is tied to the performance of the same activity by prior generations of individuals within the community, (re)emerging from the movement forward of prior joint activities (Cole & Engeström, 1993).

For a given activity, there may be multiple designs available to participants against which they can coordinate their behavior. Consider the general case of two people coordinating their efforts to pass through a doorway. There is no unique predetermined design for coordinating behavior: men open doors for women, adults for the elderly, anybody for a person in a wheelchair or a parent pushing a baby carriage, adults for children, and teenage boys never open the door for each other. Participants in an activity must agree on their assessment of the situation in order for things to proceed smoothly. Suppose Susan and Steve approach the door and Steve is older than Susan. Whether it is a case of men opening doors for women or young adults for the elderly is a matter of degree. For a joint action to run smoothly, participants must be signaling to one another how to interpret the situation; either Susan or Steve may begin to make a movement toward the door handle and begin to position their body so as to open the door for the other person.

Activities that have pre-existing designs are not completely predetermined. There are portions of the activity that are not designed and, to operate effectively, one needs to reason about those too. Removing an aspirin from a bottle is an example of emergent structure discussed by Agre (1997; p. 164). Clearly, parts of the activity are not designed. Agre's point is that there is no way to anticipate how the aspirins are arranged in the bottle before you open it. On the other hand, other parts of the activity are designed. Opening the bottle is designed; it involves pushing the cap and rotating it counter clockwise. Instructions for the design exist on the face of the cap. Even if a design for a portion of the activity does not pre-exist, the functioning of the individual's memory in support of activity may effectively impose one. If the bottle is full, an aspirin may be accessed using two fingers. If the bottle is almost empty, slowly pour one out; if more than one lands in the palm of you hand, slowly pour all but one back. In fact, these behaviors reflect the design of aspirin bottles, which in turn are subject to the constraints that emerged from prior aspirin-bottle use.

A novice cannot obtain the design of an activity purely by analysis; she must reason pragmatically (Alterman, Zito-Wolf, and Carpenter, 1998). The design can only be obtained by action, through continued practice, relying on communication between participants (even if one of the participants is in absentia). If I visit a stereo components store, each of the CD players exhibited in the showroom are likely to have a different design. In one case, switching to the next song in sequence involves pushing a button with ≫ signs on it, but for a different device the same effect is achieved by rotating a knob. For one CD player the power button is a push button and is located in the upper left hand corner of the face of the device, for another CD player it is a switch and located elsewhere. As a member of a community of actors who use such devices, I am already familiar with a large number of conventions for the interfaces for devices of this sort. Nevertheless, for a given device there is no a priori analysis that will allow me to determine exactly which conventions and in what sequence they are to be followed. There are no rules of inference (be it induction, deduction, or abduction) that will allow one to determine the exact state of affairs at the scene of the activity in using a device independent of the activity of using the device.

3. The computational model

Model construction is an important part of scientific reasoning and analytic problem-solving. The computational model has played a special role in Cognitive Science as a method for merging results and ideas across disciplines. The computational model discussed in the remainder of this paper examines the interplay of the individual and social elements of cognition. From the perspective of individual psychology, the task is to rework, in a manner consistent with the larger context, theories of cognition that resulted from a history of experimentation in indoor psychology. From the perspective of the social theories of cognition, the computational model shows how properties of the larger system can emerge from the interplay of individual cognitions. By integrating results across disciplinary boundaries, constraints and opportunities emerge that lead to general progress.

Starting with this section, the paper presents some of the details of a computational model of the emergence of convention in circumstances where there is not a ruling body of knowledge developed by prior generations of actors within the community to draw on as potential designs for cooperative and coordinated behavior.

3.1. Domain: movers-world

In the remainder of the paper, we will model and experimentally investigate everyday reasoning about conventional behavior using the computational testbed of MOVERS-

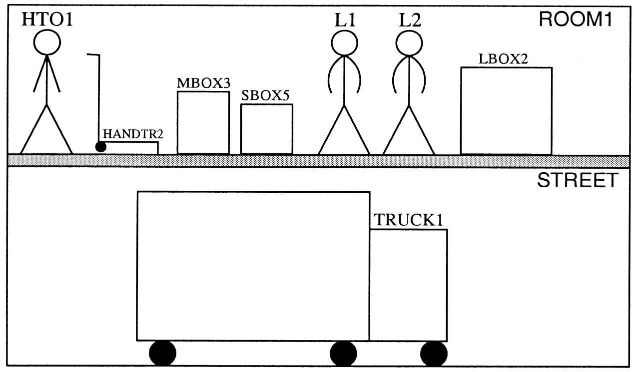

Figure 1: Sample MOVERS-WORLD problem.

WORLD. The task in MOVERS-WORLD (see Fig. 1) is to move boxes from a house onto a truck. The conventions that develop depend on specific characteristics of the participants in the joint activity and on the features of objects and artifacts about which they reason.

MOVERS-WORLD has two types of actors: hand-truck operators (such as HTO1 in Fig. 1) and lifters (L1 and L2). MOVERS-WORLD contains three types of objects: boxes, hand-trucks, and a single moving truck. Boxes come in four sizes: extra-large (XLBOX), large (LBOX), medium (MBOX) and small (SBOX). Large and extra-large boxes require two lifters to lift, and extra-large boxes are too unwieldy to be carried. Hand-trucks can hold one large or extra-large box or any combination of two smaller boxes. Hand-truck operators are not capable of handling boxes directly so loading and unloading a hand-truck requires the cooperation of at least one lifter.

The actions in MOVERS-WORLD are represented in the "minds" of the individual actors as STRIPS operators (Fikes, Hart & Nilsson, 1972). Depending on their type, different actors have different operations that they can perform, that is,

lifter: LIFT, LIFT-TOGETHER, CARRY, CARRY-TOGETHER, LOAD, LOAD-TOGETHER, UNLOAD, UNLOAD-TOGETHER, PUT-DOWN, and PUT-DOWN-TOGETHER.

hand-truck operator: PUSH-HANDTR, TILT-HANDTR, and STAND-HANDTR.

There are also general purpose operators that any actor can perform, i.e.,
MOVE, WAIT, and SIGN.

The actors do not know the types of other actors or even have an internal representation of the concept of type. Furthermore, they have no built-in knowledge about the operators for actors of other types; for example, lifters do not know initially know how the hand-truck can be utilized.

Some important features of the model are:

1. Initially actors have no knowledge of the social roles or capabilities of other actors.
2. Participants reason separately about their joint activity.

3. Ongoing communication is central to cooperation and coordination during the activity.
4. Regularities in the coordination of behavior develop with practice.
5. Memory is distributed among the individual actors; there is no central memory.

For the experiments discussed later in the paper, the community solves 420 problems. Each problem requires the actors to move from 3 to 5 boxes of various sizes to the truck. The baseline system initially takes, on average, 1026.6 simulated time steps (called "ticks'' hereafter) to solve a problem.

3.2. Coordination of behavior

An example of a joint activity in MOVERS-WORLD is L1 and L2 loading a large box onto the hand-truck.

L1, L2, and HTO1 agree that large box LBOX2 will be loaded onto the hand-truck. HTO1 must stand the hand-truck before any box can be loaded on top of it. L1 and L2 must jointly engineer the lifting and loading of LBOX2.

The joint activity of L1, L2, and HTO1 is a complex of coordination problems. Coordination of entry and exit times for each of the phases, from lifting the box to loading the box onto the hand-truck after HTO1 maneuvers it into a standing position, are jointly achieved by the participants. L1 and L2 must initiate a sequence of actions at the same time and then, step-by-step, they must execute lift-together and load-together. If they fail to achieve synchronicity, breakdowns will occur that require explicit agreements (via communication) to recoordinate the actors.

Depending on the circumstances, each of Clark's (1996; p. 64) four types of coordination devices may be in play for an activity like loading LBOX2 onto the hand-truck. When the hand-truck is standing in the same room as LBOX2, it signals the readiness of the hand-truck for loading (salience). L1 and L2 could communicate to one another when they are ready to begin lifting (explicit agreement). If HTO1 has only done this once before and, on that prior occasion, her job was to ready the hand-truck for loading, then she could assume her job in the current setting is to stand the hand-truck (precedence). Loading a large box onto the hand-truck is a regular problem of coordination among the community of actors in MOVERS-WORLD (convention).

Over time, given their understanding of the current regularities of coordination, the actors are continuously (re)learning the best ways to coordinate behavior. Actors rarely (if ever) have complete theories or agreements on the optimal way to act. Consider a problem situation where two lifters and a hand-truck operator are in a room with an extralarge box and two medium-sized ones. A reasonable way to coordinate behavior is:

1. A lifter and the hand-truck operator agree to load the hand-truck with the extralarge box.
2. The lifters lift the extra-large box together.
3. The lifters load the extra-large box onto the hand-truck.
4. In parallel:
    a. The hand-truck operator pushes the hand-truck to the street and stands it up.

b. Each lifter carries a medium-sized box to the street and loads it onto the truck.
5. A lifter and the hand-truck operator agree to load the extra-large box onto the truck.
6. The lifters unload the extra-large box from the hand-truck.
7. The lifters load the extra-large box onto the truck.

At every turn in the action, the actors are reasoning about the situation in complementary manners. Attaining this performance level of coordination between actors is an ACHIEVEMENT. This exact scenario may or may not have occurred previously, for example, the previous time there could have been another box or actor present. Initially, an individual actor has very little knowledge about the roles, skills or predilections of other actors. Another complication is that the MOVERS-WORLD actors reason and learn independently (there is no central control). Achieving this level of coordination depends on effective communication during the course of action and the existence, among participants, of a common set of expectations about points of coordination for recurrent problems of coordination.

It would be possible to engineer the domain in order to simplify the coordination problem to fit the requirements of current Artificial Intelligence planning technology. We could give each actor, at the outset, a complete specification of all the actions, attributes, and capabilities of each of the other actors. Such a design would almost reduce the planning problem for a multi-actor system to the planning problem for a single actor system. Actors would still have to communicate about intentions and commitments, for example, but an individual actor (all alone) could plan out a cooperative activity. This engineering of the domain is common practice in Artificial Intelligence. Given a problem domain, the first order of business is to reduce that problem domain to a formulation for which current AI techniques apply. But for the cognitive modeler, these kind of reductions are problematic. Giving individual actors complete knowledge of other actors capabilities and priorities may simplify the domain, but is unwarranted even upon casual examination of the relevant phenomena. Because there is no central planning architecture, MOVERS-WORLD is both a more realistic testbed for modeling conventional behavior and a challenging domain for developing next generation Artificial Intelligence technology.

4. Modeling the social interaction

The basic cycle for the community of movers-world goes as follows:

1. A problem is generated for the community of actors to solve.
2. The community of actors solves the problem.
3. Offline learning occurs.

A problem is a set of "top-level" goals and a description of the initial state of the world; for the experiments in this paper, all actors are allocated the same set of top-level goals. The overall control structure for the community of MOVERS-WORLD is not hierarchical nor is planning centralized. Actors do not share an overarching plan or reason explicitly about group concepts (Levesque, Cotten, & Nunes, 1990), rather cooperation emerges from local interactions. The capacities of other actors is not treated as a given, but this information is acquired through experience. Communication is an action, and it occurs at run-time. Each of these assumptions can be contrasted to assumptions that have been made in prior work in Distributed Artificial Intelligence and Multi-Agent Systems (Georgeff, 1983; Durfee & Lesser, 1987; Genesereth, Ginsberg & Rosenschien, 1986; Tambe, 1997; Corkill, 1979; Wilkens & Myers, 1998).

Improvement in the performance of the community occurs over several episodes of community activity. During step 2, individuals are satisfied with any solution. In step 3 the actors store into memory the portions of the behavior that they believed were essential to the solution. In future episodes, segments of the behavior are recalled and further improvements can occur. Thus, expectations about points of coordination can develop over time leading to improvements in the performance of the community.

The memory of the community of actors is distributed: individuals have access to only their private memories of their experiences. There is no off-line discussion to form a consensus on the best way to accomplish their joint goals. At issue will be the processes by which actors within the community, with sometimes overlapping and sometimes conflicting expectations, develop conventional behaviors for coordinating joint activities for regularly occurring problem situations in the domain of activity.

4.1. Communication

In MOVERS-WORLD, communication is the central mechanism for establishing and maintaining cooperation and coordination. Actors limit themselves to single requests; transmitting single requests lowers both communication costs and plan merging costs, which can be considerable for distributed, independent actors.

Two actors are said to 'cooperate' if they act or work together to achieve some common purpose and they are 'coordinated' when their individual actions are appropriately ordered to support cooperative activity. In our framework, communication is the only mechanism whereby actors can determine if they are cooperating. In other words, there are no global structures, such as blackboards (Lesser, Fennel, Erman, & Reddy, 1975), for actors to use to determine if they happen to be working on the same goal. Each MOVERS-WORLD problem-solving episode includes goals that can only be solved by cooperating actors, so communication is an essential part of the community activity.

Communication is used to attempt to establish cooperation when the set of actors working on a goal at a given time is inadequate. Cooperation is not guaranteed during communication since actors have their own decision-making strategies; even if an actor is willing to cooperate, she may be unable to do so. An actor who is unwilling or unable to assist can propose an alternative that the original requester may now contemplate adopting. Each actor presently uses the same strategy for deciding whether to cooperate: preference is given to the plan that achieves her top-level goals in the most time-efficient manner. Despite their common strategy, actors will make different decisions because of differences in their experiences.

Communication between actors in our system is similar to a telephone conversation; only two actors can participate in a single conversation. Actors do not have to be in the same location to engage in communication. One actor can call another and either establishes a connection, gets a busy signal, or the other actor does not respond. Once a connection is established, communication is handled via request and response frames. After agreeing to cooperate, an actor can opt-out at any time, without obligation to notify other actors.

A conversation between actors is made up of *requests* and *responses*. An example of a request a LIFTER might make is: “Would you help me achieve (HOLDING-TOGETHER XLBOX3).” Three possible responses to a request like this are “Yes, but you will have to wait,” “No, I don’t want to,” or “I can’t, but I was hoping you would help me achieve (ON MBOX1 HANDTR).” There are three initial request types and eighteen responses, twelve of which make an alternate request to the original caller. In practice, not all of the 234 potential sequences of request and response types are possible; for the 420 learning trials described in the empirical results section, there were 55,584 conversations comprising 67 different sequences. 53,710 conversations, in 38 different sequences, contained a single request and 1,874 conversation, in 29 sequences, contained more than one request.

4.2. Coordinating joint activity

Communication and *waiting* jointly form the mechanism whereby joint activities are coordinated in MOVERS-WORLD. Communication is used to establish agreement about points of coordination. Both communication and waiting are methods used to determine entry and exit times for the phases of a joint activity. When cooperation is first established during communication, the actors must determine how they will coordinate the cooperative activity. Sometimes, nothing need be done —for example, if two lifters are both adjacent to a ready-to-be-lifted box and both are ready to lift it. More often, though, the requester will idle for one or several time steps —for example, if the requestee lifter is not currently ready to lift the box. Another common situation is when a hand-truck operator idles after a lifter agrees to load a box onto a hand-truck. Idling is presently implemented by adding a WAIT to the beginning of her plan. While the WAIT is at the beginning of her plan, an actor is waiting for one of two events to occur: communication indicating that joint action can occur (e.g., the other lifter now indicates she is ready to act) or the completion of her request (e.g., the box appears on the hand-truck). If an actor is idle too long, she will become frustrated and inquire about the status of her request, possibly discovering the other actor has opted out. Other ways to coordinate actors are by observation (Huber & Durfee, 1995) or plan recognition (Huber & Hadley, 1997), but these techniques require that detailed knowledge about the motives and capabilities of the entire community is built into each actor to begin with.

The following two examples show trace snippets involving cooperation in unloading a box from a hand-truck and loading it onto the truck. The examples are actual output of the system (with unrelated actions trimmed), but they are illustrative, not prototypical; empirically, actors agree to cooperate less than half of the time when they have different goals. In Fig. 2, L1 happens to have the same goals as HTO1 before conversing; in Fig. 3, she does not. In both cases, HTO1 will add a WAIT operator to her plan as a result of the conversation with L1.

As Fig. 2 shows, when L1 and HTO1 have the same goals, their behavior is fairly efficient because L1 moves to the street before HTO1 makes her request. However, this kind of timing is exceptional. In this case, HTO1’s request is superfluous, but there is no way for the

```
Ticks  152 to  176: <MOVE PR24-STREET1> by L1 successful

Ticks  177 to  189: HT01 and L1 converse
  "L1, would you help me achieve (ON PR24-MBOX0 TRUCK3)?"
  "HT01, I'm already working on it!"

Ticks  190 to  224: <UNLOAD PR24-MBOX0 HANDTR3> by L1 successful

Ticks  225 to  259: <LOAD PR24-MBOX0 TRUCK3> by L1 successful
```

Figure 2: Coordination based on first-principles: Same Goals.

hand-truck operator to know this beforehand. A more common case, when the actors have different goals, is shown in Fig. 3. As mentioned above, there is no guarantee L1 will agree to help in this situation.

The responses that actors give during communication depend, in part, on how their current plan relates to the incoming request. Since wait operators represent past and potential agreements (important relationships between plans and requests), responses are strongly influenced by their presence. Wait operators are generated by agreeing to wait or through the recollection of prior behaviors (which included agreements to wait). In other words, as shown in Table 1, wait operators become part of the actor's plan either during a conversation or during off-line learning (see Section 5.1.2).

There are two variants of the WAIT operator contained in past memories of coordinated behavior. They are functionally equivalent to WAIT, but there are semantic differences that are relevant during communication. The first variant, WAIT-FOR-REQUEST, is introduced whenever an actor agreed to a request during a previous activity. It acts as a place-holder to represent when the actor **expects a request** to be made; during communication, this must be treated differently than WAIT, which is only present when the actor has an explicit agreement. The second variant, WAIT-IMPLICIT, is an optimization. A WAIT-IMPLICIT replaces a SIGN operator that would be requesting a service to be performed. For example,

```
Ticks  131 to  150: <STAND-HANDTR HANDTR3 PR24-STREET1> by HT01 successful

Ticks  151 to  163: HT01 and L1 converse
  "L1, would you help me achieve (ON PR24-MBOX0 TRUCK3)?"
  "HT01, I'll help, but you'll have to wait a bit."

Ticks  164 to  188: <MOVE PR24-STREET1> by L1 successful

Ticks  189 to  223: <UNLOAD PR24-MBOX0 HANDTR3> by L1 successful

Ticks  224 to  258: <LOAD PR24-MBOX0 TRUCK3> by L1 successful
```

Figure 3: Coordination based on first-principles: Different Goals.

Table 1
Kinds of wait operators

Told to Wait	Learn to Wait
WAIT	WAIT-FOR REQUEST WAIT-IMPLICIT

with time, the hand-truck operator can learn when to expect the lifter to load the hand-truck without explicitly being told to do so. Thus, a WAIT-IMPLICIT is an optimistic plan modification: the actor expects a service by another actor without having to ask for it.

In Fig. 4, the actors are working from plans that are derived from past joint activity and the lifter's plan contains a WAIT-FOR-REQUEST operator. Although the two actors are working from the same past interactions, L1 becomes frustrated waiting for HTO1 to make her expected explicit request. As it turns out, L1's inquiry came in the same tick that HTO1 would have made the request, so the conversation proceeds smoothly. Note that, from an efficiency stand-point, this is not better than either of the first-principles solutions. However, in this case, the outcome is more robust: it does not depend on serendipitous timing or agreeable actors. Examples of WAIT-IMPLICIT are given in Section 5.1.3, in the discussion of optimizations.

5. Modeling the individual

5.1. Memory of coordinated behavior

Throughout the practice of their joint activities, participants are learning and refining expectations about productive sequences of joint actions. Individual actors plan for new

```
Ticks  299 to  318: <STAND-HANDTR HANDTR3 PR36-STREET1> by HTO1 successful

Tick   317: L1 became frustrated since WAIT-FOR-REQUEST not satisfied

Ticks  319 to  375: HTO1 and L1 converse
  "HTO1, I was expecting a call about (ON PR36-MBOX7 TRUCK3)."
  "L1, would you help me achieve (ON PR36-MBOX7 TRUCK3)?"
  "HTO1, I've lost track of those items."
  "L1, here is that information."
  "HTO1, I'll help, but you'll have to wait a bit."

Ticks  376 to  400: <MOVE PR36-STREET1> by L1 successful

Ticks  401 to  435: <UNLOAD PR36-MBOX7 HANDTR3> by L1 successful

Ticks  436 to  470: <LOAD PR36-MBOX7 TRUCK3> by L1 successful
```

Figure 4: Coordination based on explicit expectations: Same Goals.

activities by borrowing from prior ones. The remembered success and failure of actions and decisions are the basis for improvement in behavior and the development of convention. Where initial experiences produce a set of biases, later ones refine them. New members of the community, or changes in the task environment, produce changes in the mix of regular joint behaviors that are continuously developing.

Our formulation of this issue ties learning to the functioning of the memory of the individual actors. After the community of actors solves a problem, individual actors retain in memory a description of that episode of joint behavior. This becomes the basis for the development of conventional behaviors.

Three critical features of the model we present of the individual's memory of prior coordinated behavior are:

1. Memories are derived from execution traces.
2. Memories feature points of coordination.
3. Individual actors reason independently.

There are two sets of reasons for why execution traces are the basis for memories. The first has to do with the functioning of memory retrieval. Memory retrieval is tied to surface similarity features and not higher order relations between objects (see, for example, Gick & Holyoak 1980; Gick & Holyoak 1983; Gentner 1989). By tieing memory indices to execution traces, surface similarity features are likely to be present as indices to match against cues available in the task environment. The second set of reasons reflects emergent and situative aspects of joint behavior. An execution trace for a single actor encapsulates the history of both planned and unplanned activities within the domain. Consequently, an execution trace contains more information than any of the plans of the individual actor: the trace includes the history of actor behaviors required to solve the problem. This kind of information cannot be extracted from the planning process, but it is relevant in orienting the actor towards future joint behaviors.

Joint activities are a complex of coordination problems. Remembering points of coordination predisposes the individual actor to begin to anticipate those points of coordination in future related joint activities. Individuals coordinating their behavior communicate at each coordination point in order to begin the next phase of joint activity. The points at which communication occurred between participants in the joint activity are stored as a part of the memory of prior activity. Maintaining them as a part of the individual actors' expectations about the progress of a joint activity helps to maintain common ground while keeping the joint behavior coordinated.

If two actors are attempting to coordinate their behavior by remembering prior activity, their remindings should organize their current behavior so as to reduce work as the activity unfolds. Nevertheless, actors will assess the same situation in different manners. This reflects both differences in experience between actors and the open-endedness of interpretation in general. Since remembering is dependent on the actor's assessment of the situation, actors can retrieve incompatible plans. Communication provides an opportunity to redirect retrieval to harmonize the expectations of participating actors.

There are three parts to our discussion of how memory is updated after a session of activity:

1. Extracting coordinated procedures from execution traces.
2. Optimizing coordinated procedures.
3. Storing coordinated procedures in memory.

Within Artificial Intelligence, memory-based approaches to reasoning are called *case-based reasoning* (CBR: Kolodner, 1993). Others have applied CBR technology to multiagent systems (e.g., Haynes & Sen (1998); (Ohko, Hiraki & Anzai, 1996); (NagendraPrasad, Yesser & Lander, 1995)). None of these other works focused on procedural memory, the role of points of coordination in stored procedures, or a method for learning conventions of behavior. It is important that we clarify what is meant by the term *coordinated procedures.* A coordinated procedure is not a complete specification of a joint activity. Rather it is a resource that provides a set of expectations about potential designs (points of coordination) for unfolding joint activities.

5.1.1. A simple learned coordinated procedure

The simplest plan that lifters learn that their first-principles planner does not construct is to load a box onto a hand-truck and later unload it and load it onto a truck at the behest of a hand-truck operator. This cannot be generated by the first-principles planner because initially lifters lack knowledge about hand-trucks and the behaviors of hand-truck operators. This procedure can get created from the following snippet of activity involving medium-sized box MBOX3.

First, a high-level description of the execution history

1. Hand-truck operator HTO1 asks lifter L1 to get MBOX3 onto hand-truck HANDTR2. L1 agrees and does so via lifting and loading the box. L1 next fails in an attempt to lift large box LBOX2 by herself, does nothing for a tick since she has no plan, and lifts small box SBOX5 while HTO1 tilts, pushes to the street, and stands up HANDTR2.
2. HTO1 asks L1 to get MBOX3 onto truck TRUCK1. L1 agrees, puts SBOX5 back down, moves to the street, unloads the box from the hand-truck and then loads it onto the truck.

L1 records her behavior internally, including information about active goals, cooperation agreements, states of the world, attempted actions, reasons for attempting the actions, and results of the attempts. These internal structures contain too much data to show fully, but the gist of them is clear in the following listing of the relevant portion of L1's execution trace:

```
(<agreed to achieve (ON MBOX3 HANDTR2) for HTO1>
<executed (LIFT MBOX3)>
<executed (LOAD MBOX3 HANDTR2)>
<failed to (LIFT LBOX2)>
<executed (NO-OP)>
<executed (LIFT SBOX5)>
<agreed to achieve (ON MBOX3 TRUCK1) for HTO1>
<executed (PUT-DOWN SBOX5)>
<executed (MOVE STREET)>
<executed (UNLOAD MBOX3 HANDTR2)>
<executed (LOAD MBOX3 TRUCK1)>)
```

A lot of this information is not relevant to extracting segments of coordinated behavior from the trace. Summarization and cleaning processes (described below) remove the irrelevant information. In this example, the cleaning process removes the idle step (i.e., NO-OP), the failed attempt and the two actions involving SBOX5, and the summarization process removes the LIFT, MOVE, and UNLOAD (which are reproducible) leaving:

```
(<agreed to achieve (ON MBOX3 HANDTR2) for HTO1>
<executed (LOAD MBOX3 HANDTR2)>
<agreed to achieve (ON MBOX3 TRUCK1) for HTO1>
<executed (LOAD MBOX3 TRUCK1)>)
```

The coordinated behaviors need to be represented in a more general fashion before they are stored in memory. For this example, the stored procedure would be:

```
(<WAIT-FOR-REQUEST ?ACTOR (ON ?BOX ?HANDTR)>
<LOAD ?BOX ?HANDTR>
<WAIT-FOR-REQUEST ?ACTOR (ON ?BOX ?TRUCK)>
<LOAD ?BOX ?TRUCK>)
```

5.1.2. Extracting coordinated procedures from the execution trace

Actors convert their run-time experience into coordinated procedures to be added to memory. There are six steps to this process of extracting procedures that contain expectations about points of coordination. (Further technical details can be found in Garland (2000)).

Cleaning the execution trace. Traces of execution-time activities are cleaned to remove unsuccessful behavior, which simplifies further analysis and prevents reifying past mistakes. Failed primitive actions, refused requests and idle time constitute the bulk of the trace entries removed during cleaning. This stage is a summarization process only in the broadest sense: it removes events that could not possibly have directly contributed to successfully achieving any goal; events that might have contributed are left in the trace for later analysis.

Segmenting the execution trace. The cleaned trace is next reorganized into groups of actions that are related by the goals they achieve. There are two kinds of groupings are implemented: goal-groupings and time-groupings. For each top-level goal the actor achieved, the associated goal-grouping is identified by seeing if the goal literals intersect with the operator role-fillers for each action in the cleaned trace. Time-groupings are found by looking at the actions in a time interval to see if their combined effect accomplishes a set of top-level goals; time-groupings allow the actors to learn to interleave top-level goals (see example discussed in Section 5.1.4).

Removing inefficiencies. The actor's run-time behavior will doubtlessly contain mistakes of one sort or another. To prevent reifying suboptimal behavior, it is desirable to identify and remove such inefficiencies. One could look at this step as either being a continuation of the cleaning process or as the commencement of the summarization process. The criteria for identifying inefficiencies are simple to state: actors remove primitive actions whose effects are undone (usually by herself) and communicative acts associated with such primitive actions; also, all but one out of possibly several communications about a coordination point are discarded.

Optimizing coordinated procedures. As an optional optimization, some of the coordination points may be modified. SIGNs about requests for service are changed into WAIT-IMPLICITs, reflecting an (optimistic) expectation that the request for service will be satisfied without a direct request and WAIT-FOR-REQUESTs are dropped, reflecting an (optimistic) expectation that the actor knows the right time to accomplish the request without being specifically asked. A further discussion of this process is provided below in the next subsection of the paper.

Summarizing coordinated procedures. Actions that are planner-reconstructible are removed from the segment during this step; with only a few exceptions, points of coordination are retained in summary. Two important consequences of removing reconstructible actions are to improve plan quality and reduce plan-merging effort at communication time.

Preparing summarized procedures. The low-level representation for the actions in the summarized trace segment is not suitable for reuse, so the procedure is prepared for future problem-solving episodes before storage. A straightforward change is to replace goal literals with variables. A more complicated preparation is to augment action descriptions with role-binding information that is lost when actions are removed during summarization or optimization.

Storing procedures in memory. Finally, the actor compares the prepared, summarized procedure to current case-base entries to determine if it should be added to the case-base or if a current entry should be generalized. Indexing at both storage and retrieval time is based on the goals being achieved and observable characteristics (surface features) of the setting. They are also indexed by expected points of coordination in order to facilitate retrieval during communication. At storage time, individual actors may have derived compatible procedures, but index them differently because they have different perceptions either of the start time or of the setting at the start time. Similar situation assessment problems arise at retrieval time. Discrepancies will be discovered at runtime and communication provides an opportunity for the collaborators to get in synch.

5.1.3. Optimizing coordinated procedures

Tomasello, Kruger & Ratner (1993) describe a model of cultural learning. In their model, learning depends on two levels of "intersubjectivity." First order intersubjectivity means all participants concurrently and reciprocally are able to see a situation from other actors' points of view. Second order (recursive) intersubjectivity means one actor can compare her internal point of view to (her belief about) a second actor's version of the first actor's point of view. The two kinds of intersubjectivity develop during different stages of child development.

The techniques used by actors to extract coordinated procedures from the trace depends on first order intersubjectivity. As our empirical results will demonstrate, the common viewpoint that is inherent in plans derived in this way has leverage. However, situations like

```
Ticks   77 to  111: <LOAD-TOGETHER PR34-XLBOX1 HANDTR3> by L1 and L2 successful

Ticks  162 to  181: <STAND-HANDTR HANDTR3 PR34-STREET1> by HT01 successful

Ticks  217 to  237: L2 and L1 converse
  "L1, would you help me achieve (HOLDING-TOGETHER PR34-XLBOX1)
      via (UNLOAD-TOGETHER PR34-XLBOX1 HANDTR3)?
      This is part of a plan involving you to achieve (ON PR34-XLBOX1 TRUCK3)."
  "L2, sure!  Lets get to it."

Ticks  238 to  272: <UNLOAD-TOGETHER PR34-XLBOX1 HANDTR3> by L1 and L2 successful

Ticks  273 to  290: L1 and L2 converse
  "L2, would you help me achieve (ON PR34-XLBOX1 TRUCK3)
      via (LOAD-TOGETHER PR34-XLBOX1 TRUCK3)?"
  "L1, sure!  Lets get to it."

Ticks  291 to  325: <LOAD-TOGETHER PR34-XLBOX1 TRUCK3> by L1 and L2 successful
```

Figure 5: Coordination based on implicit exceptations: Same Goals.

the one shown in Fig. 4 highlight an opportunity for improvement if the actors were capable of recursive thinking. When the actors do see the setting in the same way, and recall compatible plans from memory, they should need less explicit communication to coordinate. A hand-truck operator can hope that a lifter will load and unload the hand-truck at the appropriate times, and the lifter can hope that hand-truck operator will get the hand-truck to the street if the lifter loads a box on it.

Actor performance can be improved if some second-order intersubjectivity is added via heuristic optimizations. Two simple heuristics have been adopted in MOVERS-WORLD to implement this. Any agreements made to accomplish a request for service (as opposed to a request for joint action) are dropped instead of being converted into WAIT-FOR-REQUESTs. Reciprocally, any agreed-to requests for service are changed into implicit expectations, represented by WAIT-IMPLICITs.

Fig. 5 and Fig. 6 show scenarios where these optimizations have been made. In both of them, the box requires two lifters to handle. As the dialogues starting in ticks 217 and 273 of Fig. 5 demonstrate, not all communication has been removed; namely, communication about joint actions (which require more fine-grained coordination) has been kept. Despite this, the optimizations have led to an efficient solution by the community because the implicit expectations prevented some superfluous dialogs. In Fig. 6, the risk of the optimizations is revealed. In this case, 160 ticks have gone by without any progress being made because the lifters did not see the situation in the same way that the hand-truck operator did. Except for the delay, everything proceeds as it would have by first principles (the joint actions by L1 and L2 starting in ticks 879, 1073, and 1126 are preceded by conversations that are not shown).

```
Ticks  507 to  541: <LOAD-TOGETHER PR41-LBOX3 HANDTR3> by L1 and L2 successful

Ticks  613 to  632: <STAND-HANDTR HANDTR3 PR41-STREET1> by HTO1 successful

Tick   793: HTO1 became frustrated since WAIT-IMPLICIT not satisfied

Ticks  794 to  859: HTO1 and L2 converse
  "L2, would you help me achieve (ON PR41-LBOX3 TRUCK3)?"
  "HTO1, I'll help, but you'll have to wait a bit."

Ticks  879 to  908: <PUT-DOWN-TOGETHER PR41-XLBOX2 PR41-ROOM1> by L1 and L2 successful

Ticks  909 to  933: <MOVE PR41-STREET1> by L2 successful

Ticks  948 to 1026: L2 and L1 converse
  "L1, would you help me achieve (HOLDING-TOGETHER PR41-LBOX3)
      via (UNLOAD-TOGETHER PR41-LBOX3 HANDTR3)?
      This is part of a plan involving you to achieve (ON PR41-LBOX3 TRUCK3)."
  "L2, I'll help, but you'll have to wait a bit."

Tick  1020: HTO1 became frustrated since WAIT not satisfied

Ticks 1027 to 1039: HTO1 and L2 converse
  "L2, I'm tired of waiting.  Are you still working on (ON PR41-LBOX3 TRUCK3)?"
  "HTO1, I'm still working on it.  Chill out!"

Ticks 1027 to 1051: <MOVE PR41-STREET1> by L1 successful

Ticks 1073 to 1107: <UNLOAD-TOGETHER PR41-LBOX3 HANDTR3> by L1 and L2 successful

Ticks 1126 to 1160: <LOAD-TOGETHER PR41-LBOX3 TRUCK3> by L1 and L2 successful
```

Figure 6: Coordination based on implicit exceptations: Different Goals.

5.1.4. Example: learning to interleave goals

As mentioned previously, using execution traces as a basis for future coordinated plans allows the actors to learn plans beyond the scope of the first-principles planner (or a traditional second-order planner based on it). Another example shows how lifter L2 learns to interleave two goals. This is an useful procedure to learn since it involves little idle time and is more applicable than procedures involving more boxes. In this example, HTO1 and L1 will act as in the first example, with the box in question now an extra-large box XLBOX1 rather than MBOX3. The following are L2's plans and actions, starting four steps before HTO1 makes her first request to L1.

1. L2 creates a typical plan to get XLBOX1 onto the truck: to lift, carry and load the box jointly with L1. L1 agrees to lift the box together and they do so. L1 then agrees to carry the box to the street. However, XLBOX1 is too large to carry, even jointly, and the action (and hence rest of the plan) fails.

2. L2 creates a plan to get SBOX4 onto the truck. The plan consists of putting down XLBOX1 with L1's help and then lifting, carrying and loading SBOX4 onto the truck by herself. L2 is delayed in asking for L1's assistance because HTO1 calls first with a request to put XLBOX1 onto HANDTR2. L1 agrees to help HTO1.
3. When L2 does ask L1 to help achieve HAND-EMPTY via putting XLBOX1 down together, L1 replies that she would rather load the box together onto the hand-truck. L2's planner adapts her current plan by replacing the PUT-DOWN-TOGETHER with the appropriate LOAD-TOGETHER. The actors then load XLBOX1 onto the hand-truck. L2 continues on with her plan and loads SBOX4 onto the truck. Meanwhile, HTO1 has pushed the hand-truck to the street and L1 has agreed to get XLBOX1 onto the truck.
4. L2 constructs a plan to get large box LBOX2 onto the truck and moves back to ROOM1.
5. L2 is interrupted before attempting to lift LBOX2 by a request from L1 to help unload XLBOX1 from the hand-truck. L2 constructs the plan of moving to the street and then unloading XLBOX1. The actors do so.
6. L1 asks L2 to load XLBOX1 onto the truck and they do.

The sequence of actions L2 undertakes corresponds to six different calls to the planner. Nonetheless, L2 can extract a single coordinated procedure from her execution traces by the machinery of procedural memory (showing the original literals instead of new variables for clarity):

```
((LOAD-TOGETHER XLBOX1 HANDTR2)
(LOAD SBOX4 TRUCK1)
(LOAD-TOGETHER XLBOX1 TRUCK1))
```

5.2. Planning

MOVERS-WORLD presents a challenging planning domain. The challenge is not in handling typical planning hurdles, such as deep search trees with large binding factors; rather, MOVERS-WORLD actors must plan in an environment in which they have limited information about, and control over, other actors and the world.

The independence of MOVERS-WORLD actors yields a rich variety of run-time behaviors. At any given point in time, an actor's knowledge of the external world is her perceivable environment and a map of the world that she constructs as she goes along. Plans orient the individual actor as to how to proceed (c.f., Suchman, 1987) but, because of uncertainty in acting, plans are continuously being revised. Actors are assumed to be adaptive planners (Alterman, 1988), so if the activity does not unfold as anticipated, the actor's plan will be revised to reflect the ongoing interaction between the actor and her environment. In general, activity may not go as expected for a variety of reasons (bracketed numbers below give average frequency counts for each type of event):

1. An actor may be interrupted by a request from another actor. In this case, the actors have a conversation. During the conversation, actors may choose to suspend their current goals and plan to accommodate a request [6.3 times per problem]. Alterna-

tively, an actor may have to abandon her current goals and plan because the other actor cannot or will not provide the needed assistance [7.8 times per problem].

2. An actor may attempt a primitive action that fails because of the actor's lack of omniscience about the domain [14.1 times per problem].
3. In some cases there may be a resource conflict among actors, that is, more than one actor tries to manipulate the same object. Currently, the outcomes of these are decided at random and the 'losing' actors are delayed [3.7 times per problem].
4. Joint-actions are an all-or-nothing proposition. Unless all relevant actors participate, those that intend to attempt the joint action are delayed [$<$0.1 times per problem].
5. An actor's attempt to communicate with another actor may be delayed since the other actor's communication channel may be busy or the other actor may be refusing to answer (see Section 4.1) [18.9 times per problem].

For these reasons, actors will frequently have to adapt, replan, or suspend the current activity. When a plan is completed or abandoned, the actor selects one or more of the unmet top-level goals to actively work on. A plan is created for a set of goals by either selecting an old plan from memory or creating a plan from scratch.

The adaptation of plans that are recalled from memory is the preferred method of planning. Old plans are stored somewhat abstractly so they require some refinement before they are deployed. Our model differs from others in that extracts of execution traces, the result of multiple planning sessions occurring at various times during the activity, are stored in memory rather than the output of a single planning session, be it a plan (c.f., Fikes et al. 1972; Minton et al. 1989; Kambhampati & Hendler, 1992) or a derivational history (c.f., Carbonell, 1983; Veloso & Carbonell, 1993). Others have studied reusing plans under other conditions (Hammond, 1990), including multiple actors (Suguwara, 1995).

An actor can also create a plan from scratch using a given set of STRIPS-like operators (Fikes & Nilsson, 1971) in a hierarchical planner (c.f., Sacerdoti, 1974; Knoblock, 1990). The *situation calculus* (McCarthy, 1958; 1968) is used to represent the planner's expectations about changes in the situation that result from a given kind of action.

Each actor maintains a representation of the probability of success for different actions in different contexts. When planning from scratch, the actor uses these probabilities to guide it through the search space (c.f., Kushmerick, Hanks & Weld, 1995). This information represents some of the expectations the individual actor has about the capabilities of herself and her coparticipants. The probabilities are incrementally updated, so the actor develops more realistic expectations during the course of action. The details of the MOVERS-WORLD planning algorithm can be found in Garland (2000).

5.2.1. Probabilities

To get a feel for how estimating probabilities can help an actor behave efficiently, consider a common decision a lifter must make. A lifter can lift some boxes alone, but not all of them. If a lifter can lift the box alone, it is more sensible to do so because they do not have to spend time asking for assistance (and there is no guarantee the assistance will be given). On the other hand, if there is little chance that the lifter can handle the box on her own, it is a waste of time (and energy) to make the attempt. So, individuals should be able to recognize which of these two possibilities is more likely and act accordingly.

In MOVERS-WORLD, successfully executing an action means that after the attempt has been completed, all of the expected effects have occurred. The system determines whether actions succeed or not based upon a set of execution-time conditions unknown to the actor. The domain features that determine whether actions are possible are object size, object weight, actor strength, and item capacities. None of these are observable features of the environment. Item capacities specify how many other items and how much total weight the item can hold. The observable box features are height, width, depth and material - each measured in integers between 0 and 3. For boxes, the size is derived from the volume of the box and the weight is a linear function of the size and material. The label associated with a box includes a S, M, L or XL solely in order to make it easier for the human reader to parse the system output.

Initially, MOVERS-WORLD actors use probabilities of 50%. This steers an inexperienced lifter to try to lift boxes alone because a plan to do so will be shorter (because there is no communication needed) and thus appear more likely to succeed. So, at the beginning of a problem, lifter L1 would try to pick up a large box LBOX1 by herself. L1 would fail; the next time she wants to lift LBOX1, L1 will decide to ask for the help of another actor. This decision is based upon L1's experience interacting with that particular box.

L1's experience interacting with a particular box will not prevent her from attempting (and failing) to lift other large boxes alone. Tree structures provide a mechanism by which MOVERS-WORLD actors generalize past run-time interaction experiences, saving time and effort. The successes and failures of attempted actions are stored in a COBWEB (Fisher, 1987) tree associated with all of the observable features of the various role fillers for the action. We assume that actions might fail for reasons not explicitly considered (c.f., the *qualification problem*: McCarthy, 1977), and COBWEB can handle this noisy data. Also, COBWEB trees can be updated incrementally, which allows the actors to learn during the course of their activity. If the observable characteristics (e.g., height, width, depth, texture) of another box LBOX2 exactly match LBOX1, L1 will not attempt to lift LBOX2 alone. If the features do not match exactly and there are other experiences stored in the tree, L1's behavior depends on which experience the COBWEB classification algorithm considers the best match with the current action.

DAEDALUS (Langley & Allen, 1991) included a learning mechanism that also relied upon probabilities and COBWEB. DAEDALUS selects the operators (and partial role bindings) based upon past planning histories in order to produce a plan in more situations. The most prevalent technique for controlling behavior based on run-time interactions is reinforcement learning techniques (Kaelbling, Littman & Moore, 1996). Reinforcement learning has been successfully applied in communication-free settings (Mataric, 1992; Sen, Sekaran & Hale, 1994; Sen and Sekaran 1998) and has even been used to learn a very simple communication protocol (Yanco & Stein, 1993). Another approach to learning to control reactive behaviors can be found in Stone & Veloso (1998).

6. Experimental analysis

As the members of the community of actors become familiar with the capabilities of other members of the community and the regular problems of coordination that exist in their domain of activity, behavioral conventions begin to emerge. As actors proceed through their joint activities, expectations about behaviors of other actors are conditioned by remindings of prior episodes of coordinated behavior. Situations are never exactly the same as prior ones. Even with a growing sense that one knows how things will proceed, communication and reasoning are needed to maintain coordination between participants as the joint activity develops.

Statistics will be presented throughout this section that compare the performance of the baseline system to the performance of the system when actors are learning conventions of behaviors. By most measures, the baseline system improves over time because the actors' planners produce plans that are more likely to succeed (due to improved probability estimates). This baseline learning is significantly augmented by the addition of coordinated procedures derived from previously successful joint activities. For short, in this section, we will refer to this learning as 'learning conventions' and the cumulative procedural knowledge acquired by the community as 'procedural memory'.

6.1. Methodology

The test-bed system is written in object-oriented Common Lisp and contains 30,000 lines of source code, which produces 3MB of compiled code. The system solves both individual MOVERS-WORLD problems and sequences of them. Individual problems are constructed by randomly selecting subsets from the pool of permanent MOVERS-WORLD objects (actors, hand-trucks, and trucks) that will be active for that problem. Then a random group of boxes and locations is constructed and a list of goals involving them is generated. For these experiments, a database was created of 60 problems, whose goals were always to move all boxes to the truck. The number of boxes was uniformly distributed between 3 and 5.

The experiments are designed to control as many sources of randomness as possible in order to make comparisons meaningful. In particular, community performance can be strongly effected by the outcome of random decisions made by either the actor (e.g., selecting unmet goals to work on) or the system (e.g., determining the outcome of resource conflicts). The starting random seed value for each problem can be stored and reused, but since the order in which the actors face these decisions may vary, this is not sufficient. In addition, each randomly-decided decision point is stored together with a sequence of decisions. The first time the decision needs to be made, the first item in the list gives the result. The second time the decision needs to be made (during the course of the same activity), the second item gives the result, et cetera.

In addition to concerns about the influence of random decisions on the results of individual problems, there are concerns about the influence of problem ordering on the shapes of learning curves. It is not feasible to determine learning curves by running the system on all possible permutations of the database problems, so the system is run on seven predetermined groups of sequences. Each group of sequences is balanced in the following way: each of the

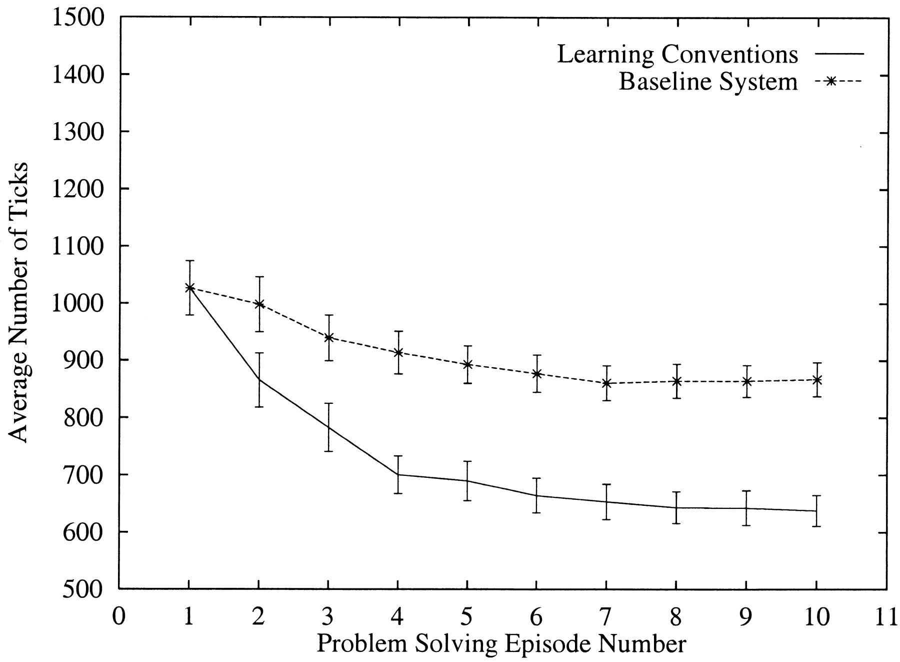

Figure 7: Overall community performance improves.

database problems occurs once as the first problem of some sequence in the group, once as the second of a different sequence in the group, et cetera. So each data point shown is the result of solving each of the 60 database problems seven times, using different sets of seeds and decisions each time; these 420 runs were repeated for both the baseline system and when the actors were learning conventions.

6.2. Overall performance characteristics

As the community of actors gets better at coordinating their behavior for regularly occurring problems of coordination, one would expect to see some general improvements in the performance of the community.

There are many ways to measure the performance of the community, such as the number of primitive actions attempted and the number conversations that occur. The best overall measure of community effort, however, is the number of ticks that transpire during the course of the community solving the problem. This measure of simulated time includes both action and communication effort, in addition to time when the actors are idle for one reason or another. For example, an actor will idle (technically, they execute a NO-OP) because their planner has failed to generate a plan [the number of times this occurs, averaged over the 420 runs, ranges from 1.9 to 3.7 times per problem].

Fig. 7 shows improvements in group performance with and without actors learning from prior experience of coordinated behavior. The 99% confidence interval for each data point is

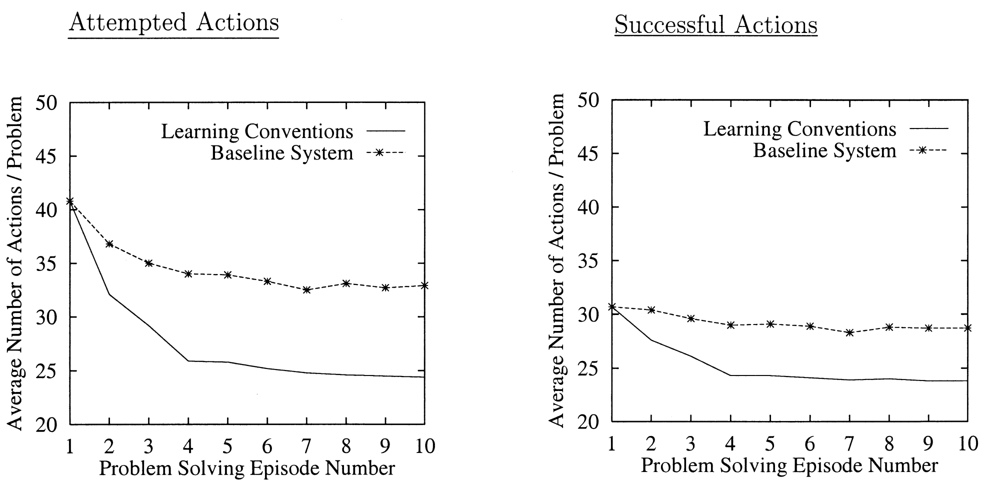

Figure 8: The number of primitive actions is decreasing.

displayed as well. The average number of ticks required to solve these problems before learning conventions was 1026.4, with a standard error of 18.5. Even without procedural memory, there is improvement in group performance; these improvements result from increasingly accurate probability estimates. The addition of procedural memory leads to significant improvement; the average number of ticks drops fairly steadily, ending at 638.1, a 26.4% improvement over the baseline system. Improvement along each curve after the fourth problem-solving episode were not statistically significant, but they do suggest that the community continues to modestly improve at the task.

Overall system performance of an AI system is often measured by CPU usage. However, CPU usage is an inappropriate measure of community performance in our domain because we are primarily interested in improving the community's runtime behavior, not the speed at which they plan (or retrieve plans from memory).

This disclaimer does not hide an ugly result – CPU time is less when actors learn conventions.

6.3. Primitive actions

Fig. 8 shows that the improved runtime performance of the community is a direct result of the fact that the planner produces plans that are either more efficient (e.g., interleaves goals), more likely to be successful, or both. The left side of the figure tracks the number of actions attempted (whether successful or not) by the actors over the course of solving a problem. The right side focuses on just the number of successful actions. Learning conventions leads to significantly fewer actions, both attempted and successful. Standard deviations for all points shown are less than 0.82 and, again, there is no statistically significant decrease along the tail of the curves.

By focusing on the performance of the baseline system, we can see that, over time, the improved probability estimates lead the individual actor to produce plans that require fewer

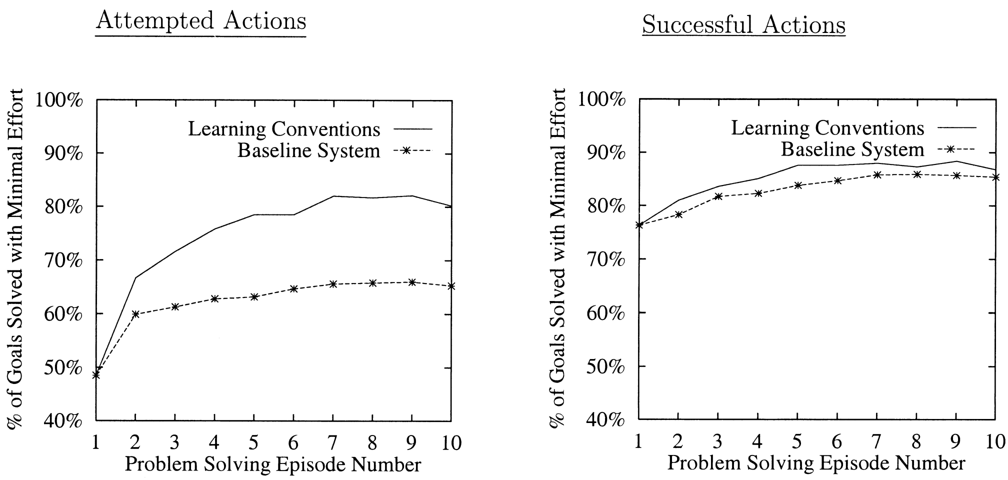

Figure 9: Approaching minimal action effort.

attempted actions. However, this is not translated into a reduction in the number of successful actions needed to solve the problems. So the reduction in the number of ticks for the baseline system reflects the fact that actors find the appropriate kinds of first-principles plans earlier

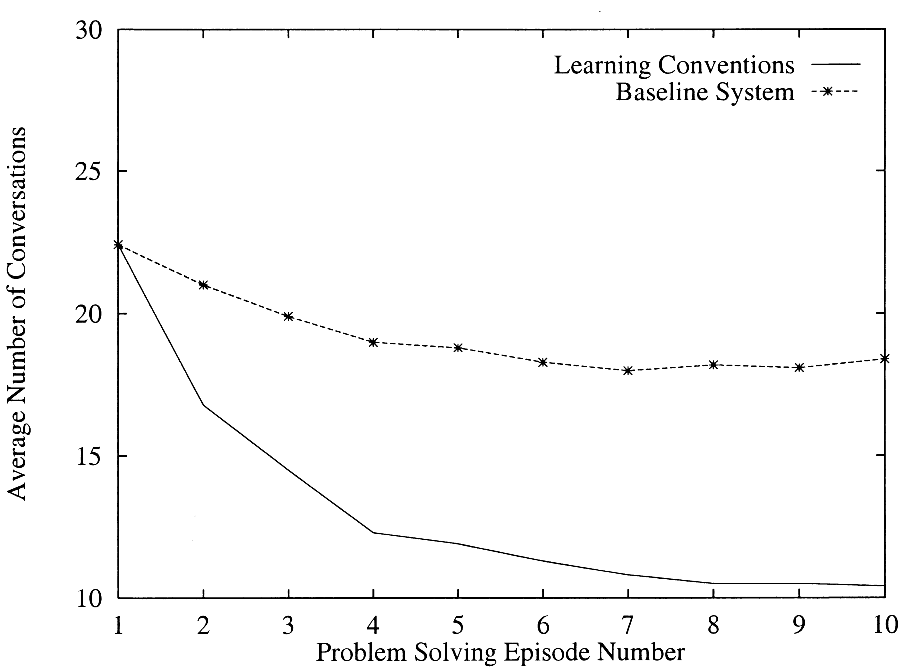

Figure 10: Community communication decreases.

in the course of the activity. On the other hand, procedural memory allows the actors to solve the problems using fewer successful actions. Clearly, the actors are guided by higher caliber plans.

Fig. 8 takes a macroscopic view, measuring the action effort of the community to solve the entire set of top-level goals. Alternatively, we can myopically measure how efficiently the actors are solving each top-level goal in isolation. For a given goal, we can compute the number of actions, either attempted or successful, that the actors undertook to achieve the goal. This can be compared to the fewest possible given the initial configuration of the problem. For example, a small, clear box inside the house requires a minimum of three actions to get it onto the truck: LIFT, CARRY and LOAD. Then, for all problems and all common top-level goals, the frequency with which the community undertakes the minimal number of actions can be computed. The results are given in Fig. 9. With procedural memory, the actors in MOVERS-WORLD are attempting the minimal number of actions more frequently (80.2% of the time) than in the baseline system (65.3% of the time). The actors also solved problems using the minimal number of successful actions more frequently when they learned conventions.

6.4. Communication

Fig. 10 shows that learning conventions in MOVERS-WORLD reduces the amount of runtime communication. As individual actors get better at guessing what kinds of joint activities work and how they will unfold, the amount of runtime communication needed to achieve their joint goals is decreasing.

This reflects possibly two things: the number of requests is decreasing and/or the percentage of agreed-to requests are increasing. The number of requests is decreasing either because actors are making fewer requests that are rejected, or because the actors are able to anticipate a request or action on the part of another actor without having to directly communicate. Our experimental data show that by the tenth problem solving episode, with the use of procedural memory, the number of requests from one actor to another has decreased but the percentage of agreed-to requests has increased, that is,

	Requests	**Percentage agreed-to**
Initial behavior	19.6	73.7
Baseline system	17.4	62.7
Learning conventions	8.7	77.9

A natural question to ask is whether the improvement in runtime performance comes at a high price in increased planner effort. The data show (see Fig. 11) that there is a decrease in the amount of search effort the planner requires in order to instantiate local role-binding variables. There is also a dramatic reduction in the number of planning search nodes expanded during calls to the planner, both compared to the initial behavior and the baseline learning system. In other words, the development of conventions is a ‘win-win’ situation.

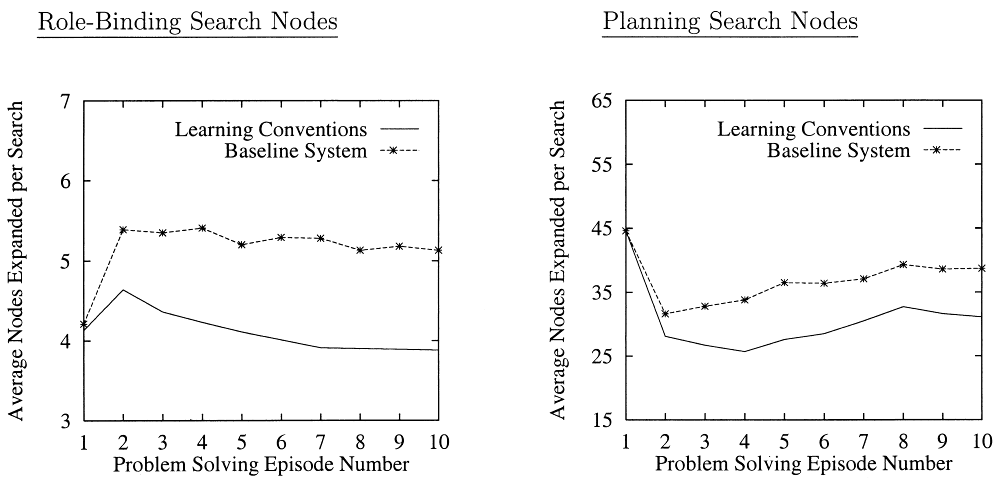

Figure 11. Planning effort decreases when actors learn conventions.

6.5. How is memory for prior coordinated activity improving performance?

The analysis presented so far confirms that the participants are getting better at coordinating their behavior when learning conventions. Several measures of external behavior reflect improvement. Overall communication and action effort are reduced; actors attempt and succeed at finding the optimal solution for moving an individual box more frequently; and the number of requests between actors is decreasing, while the percentage of agreed-to requests is increasing.

Why is the external behavior of the community improving? Our results show that learning conventions improves performance by predisposing actors to approach the problems in related ways.

We track conversations in order to measure when two actors are assessing a situation in related ways. One indication the two actors are assessing the situation in similar manners is that the listener is already working on the same goals as the requester. A stronger indication is when the listener is both working on the same goals and the listener's plan already includes a coordination point corresponding to the request. More technically, the listener's plan is considered related to the incoming request if any of the following are true:

1. The incoming request is a request for a service to be provided and the plan contains either
   a. Communication about the same request; or
   b. A wait operator for the same request.
2. The incoming request is a request for a joint action and the plan contains either:
   a. Communication about the same action; or
   b. A wait operator for the same action; or
   c. The same action.

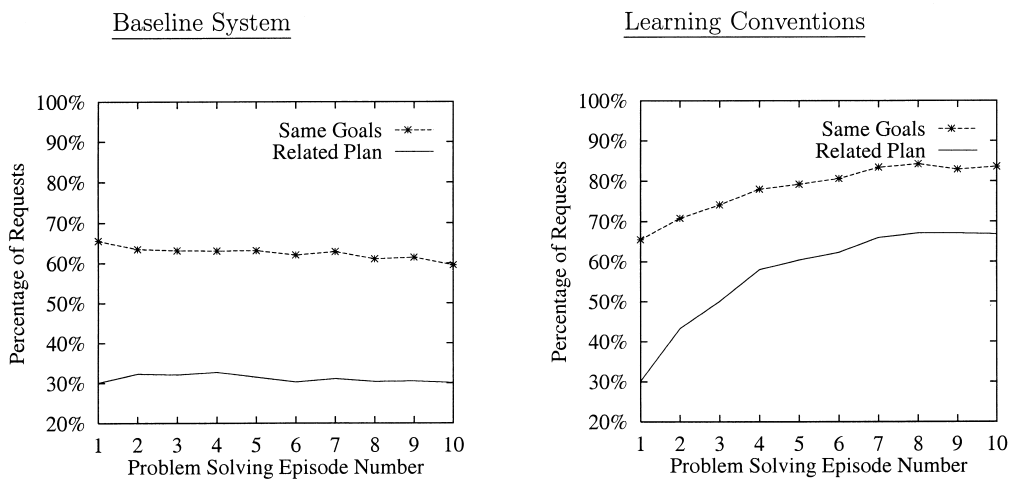

Figure 12: Measuring how conventions lead to related approaches.

The ideal situation would be that when one actor makes a request of another, the two actors always are working on the same goals and have plans containing the same coordination points.

Fig. 12 measures how frequently two conversing actors have similar assessments of the situation. The left-hand side of the figure measures the performance for the baseline system. For 66% of the requests, the listener has the same goals and 30% of the time she has a related plan. The right hand side of the figure measures performance when the actors use their memory of prior coordinated behavior to inform their activity. Here we see improvements in both statistics. More significantly, the curves are converging. In other words, the actors are learning to assess the same situation in compatible manners by extracting plans with related points of coordination to accomplish the same goals.

Despite the fact that actors are converging on individual coordination points, they do not converge on an identical mental structure for representing conventional behavior. We measured the overlap in the mental structures of independent actors in two ways. One way compares pairs of case-base entries from different actors to see how frequently they have the same plan. This entails having the same number of actions, each of which has the same action description (modulo the internal variable names). Because of their differing operator sets, lifters and hand-truck operators never have the same plan. Even two lifters do not converge on the same plan: less than 11% of the plans in memory are isomorphic. A stricter measure, which requires that the entries are stored under the same contextual indices, never breaks 3.2%.

6.6. Effect of optimizations

To demonstrate that the results in the section are not dependent on the optimizations that MOVERS-WORLD actors perform, we reran the system without them. Fig. 13 shows how

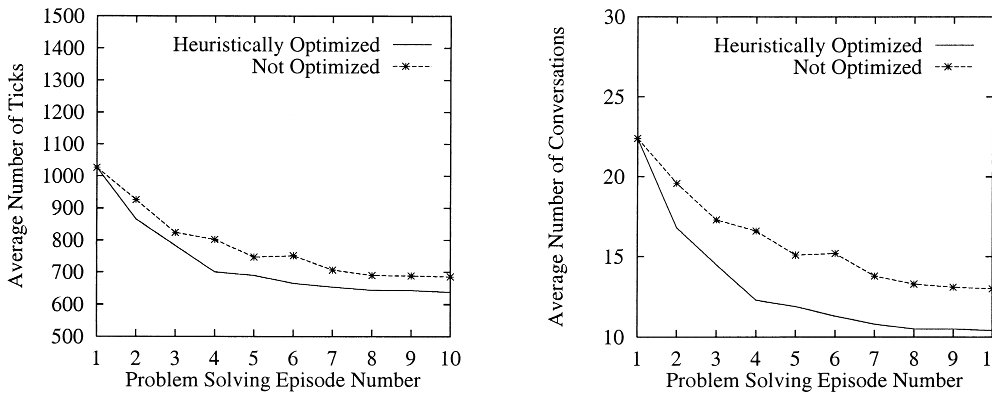

Figure 13: Measuring the impact of optimizations on ticks and conversions.

two performance measures compare. Learning is faster and performance slightly better when the optimizations are performed, but there is not a great deal of difference. Other statistics show similar relationships. Thus, the credit for the improvement in community performance lies primarily with remembering in general and not with the optimizations.

7. Discussion

7.1. Combining two views of cognition

Within Cognitive Science, a framework for the study of cognition that integrates social theories of cognition with cognitive science models of human information processing has been explicitly advocated by Hutchins (1995b) and Greeno (1998). Hutchins (1995a; 1995b; 1993) argues that cognition is distributed among the artifacts and individuals who participate in culturally and historically conditioned ongoing practices. These larger socio-technical systems have cognitive properties that cannot be reduced to the cognitive properties of individual persons. This larger unit of analysis provides a basis for the study of cognition (Hutchins, 1995b; p. 266):

> In this paper, I will attempt to show that the classical cognitive science approach can be applied with little modification to a unit of analysis that is larger than a person. One can still ask the same questions of a larger, socio-technical system that one would ask of an individual. That is, we wish to characterize the behavioral properties of the unit of analysis in terms of the structure and the processing of presentations that are internal to the system. With the new unit of analysis many of the representations can be observed directly, so in some respects, this may be a much easier task than trying to determine the processes internal to the individual that account for the individual's behavior. Posing questions in this way reveals how systems that are larger than an individual may have cognitive properties in their own right that cannot be reduced to the cognitive properties of individual persons. If the unit of analysis is a single individual, cognitive scientists can only infer internal states of the actor

> from external behaviors, while with larger systems it is possible to directly observe some of the internal states of the larger system.

Greeno (1998) advocates a program for research that combines standard cognitive science with the research and methods of work on situated activity that have come out of the social sciences. For Greeno, the key is to analyze information structures in socially organized activities (p. 6):

> Although these lines of research —the study of individual cognition and of socially organized interaction —both provide important scientific knowledge and understanding, they have developed in relative isolation from each other. Cognitive science analyzes structures of the informational contents of activity, but has little to say about the mutual interactions that people have with each other and with the material and technological resources of their environments. Interaction studies analyze patterns of coordination of activity but have little to say about the informational contents of interaction that are involved in achieving task goals and functions.

Greeno recommends that both cognitive science and situative strategies of research be investigated 'vigorously'.

The computational model of conventional behaviors that is developed in this paper fits into Greeno's proposed program of research. It details the cognition of individual members of a community of actors participating in joint activities. Individuals push through joint activities by relying on their memories of prior activities. Over time, conventions of behavior in joint activities begin to develop. But these conventions of behavior emerge only during the course of activity and are subject to the constraints of the larger context.

From the perspective of individual psychology, the computational model presented in the paper depicted everyday reasoning about conventional behavior and joint activity as a memory-based process tied to pragmatic action. The expectation that certain points of coordination would develop during the course of action framed the individuals reasoning about joint behavior. From the perspective of the distribution of cognition, the computational model showed how the social interaction circumscribed and simplified the reasoning processes.

In general, a computational model is a *description* of cognitive processes and not a *demonstrative simulation* (Greeno & Moore, 1993: p. 56). Two issues will suffice to illustrate this point. One issue concerns the selection of mechanisms to use for modeling the reasoning of the individual actor. For example, early Artificial Intelligence methods used explicit structural representations (e.g., frames: Minsky, 1975) and semantic networks (Quillian, 1968) for depicting *schemata* (Bartlett, 1932) as a part of memory. Within the parallel distributed processing movement (see Rumelhart & McCelland, 1986), the representation of the structure of memory became more distributed and moved closer to approximating what is known about how networks of neurons within the brain 'compute'. For the cognitive modeler, the trade-off between using one or the other machinery as a part of his/her model is complicated. Neural nets may better approximate how the brain works, but as a practical matter, the current implementations of neural network computation are better for modeling low level perception than they are for characterizing larger fragments of phenomena such as everyday activity. The computational modeler can use explicit structural representations or rules for this purpose, but the caveat is that relevant features of the structured representations and rules must be eventually replaced by models that better approximate the brain activity. One could try to advocate a position that all high level theorizing is irrelevant to the eventual brain theory of behavior that will emerge, but it is hard to understand the force of this argument. In general, it is well understood that top-down filtering is a powerful problem-solving tool. In the case of combining the top-down theories of cognitive science and the situative view of cognition with the brain theories of the neuroscience community, evidence fails to support a radical neuron doctrine that would discount modeling from the top-down (Gold & Stoljar, 1999).

Another sticky issue concerns the level at which to analyze and represent the phenomena. The MW model represents the activities of the participants at a fairly course grain. Take a joint action like CARRY-TOGETHER. Actors can, and do, reason about their activity at that level. A finer grain analysis would feature the reasoning that is done as the carrying together unfolds. How is the path determined and how are obstructions managed? What happens when one (or both) of the actors needs to change her grip? How does the passage through a doorway proceed? At this finer grain level of analysis, there are continual adjustments that are made by the participants in order to wend their way through the activity. The decision to model at one level or another reflects a confluence of issues. A critical factor is that each finer grain level of analysis introduces additional complexity to the computational model. The level at which MW was modeled entailed the production of 30,000 lines of LISP source code. An even finer grain analysis would have required larger amounts of code, and there is no reason to believe that the increase in code would have been linear rather than exponential. For example, at any granularity of analysis for the MW domain, one could talk about the mental parts and the action parts. It is a hotly disputed issues whether they are in fact separable. The differences of opinion pivot over claims about internal representations, interaction, and emergence. Some of these issues are addressed in the MW model, others are not. A finer grain analysis would have forced a careful examination of the constellation of issues concerning internal representations, affordances, and perception (e.g., Clancey, 1993), but the resulting additional complexity in the modeling task would have prevented a summation on the point of convention.

Each refinement in the analysis introduces additional complexity in the programming task. Consequently, the computational modeler, at some point, needs to decide which issues should be pursued and to what depth. One way to think about a computational cognitive model is that it is a summary with an ‘attitude’. It is a summary because there is too much ground to cover. It has an attitude because the model that is produced is an interpretation and characterization of the relevant phenomena.

A final point about cognitive modeling concerns its relation to Artificial Intelligence (AI) programs. At one point the framework for the study of AI was assumed to be cognitive. Nowadays, the relationship between AI and Cognitive Science is somewhat more complicated. Not all human methods are optimal methods for computers. Not all AI programs are cognitive models (e.g., Deep Blue). The AI researcher wants to achieve engineering successes, and the computational cognitive modeler wants to depict the relevant aspects of the experimental and ethnographic evidence available. An extended project can include parts that are either AI or Cognitive, and the arguments for each kind of feature differ. Nevertheless there can be confusions and conflicts between the system as a model that reflects results and data from the interdisciplinary communities and the system as a solution to an engineering problem.

There are a lot of cognitive constraints on how such a model could be put together, and each constraint introduces new complexity into the programming task. The more we pay attention to the cognitive constraints, the harder it is to build the model, as the constraints work to rule out many simplifying or closed-world assumptions or programming decisions. Every time you give up a closed-world assumption and replace it with a story that is based on psychological experimentation or ethnographic study, the models gets harder to build. One can engineer a solution to find the optimal solution to a given closed-world assumption (the AI problem), but for the cognitive scientist the task is to model the relevant phenomena in a manner consistent with existing interdisciplinary data and theory.

7.2. Where and when do conventions exist?

In MOVERS-WORLD, memory is organized around points of coordination within the progress of a joint activity. This principal of organization assumes that individual actors are trying to converge on an ideal structure for the activity. The analysis of the data shows that with the addition of this expectation (not its completion), reductions in the number of primitive actions taken, communication costs, and plan effort are all achieved as the overall performance of the community continues to improve. Section 6 shows that, given this principal of organization, the machinery of individual memory can convert raw common experiences into improvements in the performance of joint behavior. The analysis of the experimental data of MW does not, however, provide evidence that there ever exists a unique or ideal point of convergence in the mind of each of the participants (see Section 6.6). Each memory provides an orientation, a set of expectations, about how the current activity could proceed. Some recollections may be better than others, but none of the individual memories have any special status. In other words, improvements in coordination for practiced joint activities are achieved, even though no ideal or unique script for a conventional behavior is determined.

Perhaps a unique idealization of a conventional behavior exists in an external representation. For example, in the usage of an air telephone there exists, in the form of instructions, an external representation of the conventional behaviors that the individual actor must produce in order to use the device. Considerable work is necessary to achieve the behavior dictated by the instructions (Suchman, 1987; Agre & Chapman, 1990). An icon of a credit card informs the air telephone user of the method of payment, but to achieve the conventional behavior of inserting the credit card so it can be read necessitates a great deal of work (Alterman, Zito-Wolf & Carpenter, 1991; 1998).

Reconsider the case of Dick and Jane coordinating their efforts in order to achieve joint passage through a doorway. What the model suggests is that as Dick and Jane continue to practice their joint activity, they both have the expectation that an ideal and unique script for the convention will eventually emerge. Because the recall of prior joint activities orients future behaviors, the expectation of conventional behavior shapes performance. Nevertheless, as independent actors, Dick and Jane do not converge on an ideal script; effective performance will continue to require online adjustments and cooperation in order to keep on-track and in-synch. Even if the wheelchair came with a set of instructions that describe a conventional behavior for one person to aid a second person in a wheelchair at passing through a closed doorway, it would still require considerable work on Dick and Jane's part to achieve their common goal.

To recapitulate, there is no ideal internal script that represents the convention. An external representation of the conventional behavior requires work, cooperation, and coordination of effort. Whatever structure for the conventional behavior that exists prior to the joint activity at best approximates a behavior of convention. But then where, or perhaps when, is the convention?

Research studies in ethnomethodology (Garfinkel, 1967) and situated activity (Suchman, 1987; Lave, 1988; Agre, 1997) report that the structure for activity is, in part, a product of the activity. Perhaps this is also the case for behaviors that are conventional, that is, the structure of convention is at least partially a product of activity. In the case of Dick and Jane, there is no analysis that will anticipate the coincidence of details that confront them as they wend their way through the activity. By means of a program of combining expectations with the specifics of the situation that confronts them, Dick and Jane are able to coordinate their behavior. Although there are prior structures for the activity that informs participant behavior, additional structure arises during the course of the activity.

An example of a structure that could emerge during a joint activity is a *shared plan* (Grosz & Sidner, 1990). From the perspective of collaboration, each occasion of joint activity is managed through the ongoing construction of a shared plan (Grosz & Kraus, 1996; Grosz & Kraus, 1999). Shared plans may be partial and can be revised as the activity continues. For a group of collaborators to have a full shared plan there must be a mutual belief that they are committed to the success of the collaboration and a mutual belief of the need to accomplish each of the subactions upon which it depends. A key idea is that even when they only have a partial shared plan to achieve a given subaction — collaborators only partially know what they are going to do —they always have a full shared plan to work out the details of the partial shared plan. A unique shared plan is constructed on each occasion of joint activity.

Not all conventional activities are collaborations. When crossing the street at busy intersection in San Francisco, pedestrians are moving in both directions, they must cross the street in the allotted time while avoiding bumping into each other and staying within the crosswalk. There are conventions for these kinds of activity, but it would be odd to claim the participants were either collaborating or had a shared plan. Where collaborations are occurring, the structure of the conventional behavior bears some relation to a shared plan. When Ed and Joan start living together they have not worked out a convention for cleaning house before company comes over, but over time they might. If there is a shared plan for the *n*th time they perform this activity, there was also one for the first time they did it. This points to the critical difference between a shared plan and a convention. A shared plan is a construct that is tied to a single episode of joint activity. A convention is a construct that is tied to the history of related joint activities; it develops with a social practice. In other words, if the unit of measure is a single episode of joint activity, the shared plan is a synchronic analysis of the behavior of the participants and conventions are a diachronic one. Convention is measured by a reduction in the amount of work needed by participants in a joint activity to achieve a common goal. Over time, through practice at working at a recurring problem of coordination, certain points of coordination become expected, a design for the activity develops, and a convention becomes a part of the initial common ground. The emergence of convention coincides with the development over time of a home task environment for coordinating behaviors. A shared plan realizes a conventional behavior only after a community of actors have worked together, over many episodes of joint activity, to achieve a given goal in a given context. The shared plan reflects the work participants do in performing a behavioral convention on a given occasion, but the same convention may be invoked by different shared plans on different occasions.

To summarize: When do conventions exist? They exist after a community of actors has had the opportunity to interact for awhile. Where do conventions exist? They exist in the predisposition of individual actors that develops from their shared practice and is contingent on historically conditioned constraints and affordances of their shared task environment.

7.3. Everyday reasoning

The coordination of joint activity is the key to everyday behavior, and reasoning about coordination is the task of everyday reasoning.

One possibility is that "everyday reasoning" is another form of analytic/scientific reasoning. Scientific reasoners do collaborate, and it improves their performance. Okada & Simon (1997) model two students engaged in a collaborative scientific discovery task. Their evidence argues that the collaborative unit A & B performs on average better than A or B working individually. Their model of the data shows that (p. 139) "A partner's questions, requests, critiques toward one's hypothesis and/or justification enhance further search in the hypothesis space and the experiment space." They do not, however, model how the students stay coordinated during their collaboration. Whose turn is it to speak? What does my coparticipant mean when she points to the figure on the paper and speaks the words . . . ? At this level, the discovery collaboration is "An iterative cycle of displaying, confirming, and repairing situation actions" (Roschelle 1992; p. 237). Is a locally and interactionally managed form of group decision-making that is administered by the participants during their collaboration (Sachs, Schegloff & Jefferson, 1974) the same thing as a scientific reasoning? Are the discovery participants managing their interaction using scientific reasoning?

With scientific and analytic reasoning, the task is to build an abstract model that characterizes a given set of phenomena (data). Both analytic reasoning and scientific reasoning are modeled in cognitive science as heuristic search (VanLehn, 1989). Pure analysis (solving puzzles, doing proofs) is heuristic search (Newell & Simon, 1972) and the assumption is that an analytic solution exists to achieve a goal state. The scientific reasoner searches the space of possible models to find a good fit for the relevant data; the problem is to construct a single consistent model that covers all the relevant data and only the relevant data.

With everyday activity, the goal state has a feature of arbitrariness. Analysis will take the individual only so far. Whatever 'analysis' the individual makes about the predicament in which she finds herself falls short of uniquely determining the course of joint action. While Dick and Jane are achieving coordination as they move through the doorway, there are points at which they cannot predict with sufficient certainty —the coordination points —what the other will do. And they need to, or the joint action will fail. And that is the problem of "everyday reasoning."

There is no way to analytically determine what convention is in play. One cannot guess what the other participants are going to do, and there is no way to proceed unless you do. So the participants signal back and forth to stay on course, righting themselves with even more social interaction when breakdowns in coordination occur. It is the interactive part —the social part —that drives the system. The best that the individual actor can hope for is getting better at expecting certain points of coordination to occur. A given is that the expectations of coparticipants will differ. No doubt heuristic search is also occurring to prepare for the interaction, but it is also greatly circumscribed and reduced by the social interaction. For another discussion of everyday reasoning versus analytic reasoning, see Alterman (1999).

7.4. Are two minds reducible to one?

What is the practicing cognitive scientist to make of social theories of cognition? The tradition in Cognitive Science has been to choose a single mind working alone as the basic unit of analysis. The skeptical cognitive scientist might agree that there are social, cultural, historic, and interactionist aspects to everyday reasoning about the coordination of activity, but argue that the analysis can be framed in terms of the inner working of the individual mind. If one could show that the social level is decomposable into lower levels that involved the reasoning of one mind, then the case would be made. Can it be shown that two minds are reducible to one?

For the sake of argument, suppose the world can be divided into two parts: an outside and an inside. Outside is the external world and the task environment, and inside is a symbolic representation of the relevant aspects of the task environment. Cognition works inside a closed room from internal representations, analyzing their structure, rationally constructing plans for future behaviors; it is framed as heuristic search in a problem space representation of the external world. Cognition and behavior function like a sewing machine: inside is cognition; outside is behavior; inside is structure; outside is action. Over repeated cycles, behavior takes shape, but the intelligence in the behavior is only occurring within the closed room "sandwiched" in-between moments of contact with the external world.

If two minds are reducible to one, then the structure of behavior is only inside or, at a minimum, the creation of structure depends only on what's inside. Is structure only inside? No, structure is both inside (e.g., a plan) and outside (e.g., instructions). Does all the structure of behavior originate from the inside? Or does some of it require a social interaction to initiate creation? One could argue that the structural elements of language (Chomsky, 1980) and thought (Fodor, 1983) are rooted in the biological. One could also argue over whether external structure at one point or another has an internal form (see Cognitive Science, 1993). But both of these are beside the point. The issue here pivots over whether there exists structure for behavior whose creation depends on a social interaction. If none exist, then two minds are reducible to one.

As we have shown in this paper, in the case of joint activities and conventional behaviors, some of the structure of behavior clearly originates from the social interaction. The participants work to stay coordinated in order to simplify and improve performance. The way they stay coordinated is to exchange information about their expectations about the structure of their joint behavior. When Dick and Jane engage in a joint activity, whatever predispositions exist, in the form of expected point of coordination, only abstractly characterize the structure of their joint behavior. Additional structure for the behavior (the collection of these exchanges of information) emerges from their joint activity, and these are external representations. The participants co-construct a design for conventional behaviors. The design that emerges exists only as a product of social interaction; it does not exist without the interaction.

Even in the case of individual actions that do not directly involve other actors, the structure for the activity partially originates in an external structure that was provided by another actor, or the same actor at an earlier time. Most —all? —individual behaviors are mediated by tools and external signs. Learning to do new activities depends on instruction. Instruction can come from many sources: another person, written instruction, or design of the task environment. All of these, either directly or indirectly, depend on input from another person who is already familiar with the structure of the activity. This dynamic is the basis of cultural history and learning (e.g., Vygotsky, 1978; Cole, 1996). Examples of models of cultural learning that depend on a social interaction include: Hutchins (1995a), Lave & Wenger (1991), Alterman et al. (1998), and Tomasello et al. (1993).

Even when one's behavior in using a given mediating artifact has become more routine, it is still dependent on the design of the task environment, which is an external representation provided by another actor. Take an action like "dialing" a number on a cordless touchtone phone. The individual actor may know the number she is going to dial and she knows she will grip the phone in her right hand and press the telephone number buttons using her right thumb. So, some of the structure of her behavior is internal. But clearly not all. She may know the layout of the touchpad, but her hand will be positioned differently on each occasion, so some of the structure of her behavior, the control of the "dialing" is structured by visual information available as she performs the act. It is the *uncertainty* of the relation of the hand to the cordless phone, as she holds it, which makes a strategy of using external representations available in the task environment to guide behavior advisable. Another way to state this advantage is that distributing the structure of the behavior between the design of the task environment and the internalized procedures of the individual make everyday reasoning about behavior more reliable and robust.

Learning and uncertainty are two reasons that the structure of behavior can originate in the external. The reliability of memory is a third.

Two performance characteristics of long term human memory are retention and accessibility. The retention of information in long term memory takes practice. Even for information that is retained, recall may be slow or fail. During the flow of activity, any slow down or failure to recall relevant information can cause breakdowns. Associating controls with burners, is easy to do for someone who regularly cooks at home, but that information is easily confirmed and more readily accessible, if the actor looks at the labeling of the controls as she proceeds with her activity. But again, that means reasoning about everyday behavior is dependent on external representations of the structure of the requisite behavior — without them the behavior will continuously break down. One could argue that before the individual acts, the relevant design information is encoded within an internal representation. But that misses an important point, which is that the rerepresentation within the mind of the individual is a derivative of the external representation provided by another actor.

8. Concluding remarks

Reusing and extending conventional methods for coordinating behavior is the task of *everyday reasoning* and the subject of this paper. The framework we developed combines social theories of cognition with human information processing models that have been developed within Cognitive Science.

From the perspective of a social theory of cognition

- The social interaction among participants circumscribes and simplifies the reasoning processes. The participants agree to points of coordination as they proceed. The historic elements of the social interaction between actors within a community cannot be factored out of the analysis of conventional behavior.

From the perspective of individual psychology:

- Reasoning about everyday joint behavior is a memory-based process tied to pragmatic action. The predisposition for certain points of coordination to develop during the course of activity, and the assumption that other actors will share those predispositions, frame the individual's behavior.

Conventional behaviors develop as a part of the social practice within a community of actors and are partially emergent. Different actors bring to bear different expectations, knowledge, and experience. As the actors proceed with their joint activities, gaps in behavior are filled in by reasoning about historically conditioned constraints, affordances, and patterns of coordination. During the give-and-take of activity, participants reason with a bias that at one point or another their expectations about points of coordination will be met. There is also the understanding that because of uncertainty, interruptions, and numerous other opportunities to get off-track and out-of-synch, the participants must work continuously and jointly to achieve conventional coordination. The expected points of coordination collectively form a *design for the activity,* but not a complete specification of structure of the convention activity, which emerges uniquely on each occasion of recurrent activity.

The computational model we presented details the emergence of convention in circumstances where there is no ruling body of knowledge developed by prior generations of actors within the community to guide behavior. The example domain is a group of actors who are part of a moving company. Their job is to move boxes from a house into a truck. With practice, individuals within the community begin to converge on a set of conventions for behavior that match the regularly occurring problems of coordination in the domain of activity. One feature of the model is that the mechanisms for improving behavior are tied to the memory function of individual actors. Another important feature of the model is that the community improves its performance despite the fact that individuals reason independently about their experiences. A large number of computational experiments were conducted. The empirical evidence supports the theoretical position that was developed on conventional behavior. Convention is measured as the reduction in the amount of work needed by participants in a joint activity to achieve common goals. The development of convention reduces communication, planning costs, and the number of primitive actions needed to achieve common goals. There is no unique internal structure in the mind of all participants that represents a conventional behavior.

Acknowledgement

The authors would like to thank the following people for their comments on an earlier draft of this paper: James Greeno, Edwin Hutchins and two anonymous reviewers. This work was supported by ONR (N00014–96-1–0440). Additional funding was provided by NSF (ISI-9634102).

References

Agre, P. (1997). *Computation and Human Experience.* Cambridge University Press.

Agre, P. E., & Chapman, D. (1987). Pengi: An implementation of a theory of activity. In *Proceedings of the Sixth National Conference on Artificial Intelligence,* pages 268–272.

Agre, P. E., & Chapman, D. (1990). What are plans for? In P. Maes, editor, *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back,* (pp. 17–34). Cambridge: MIT Press.

Alterman, R. (1988). Adaptive planning. *Cognitive Science*, 12, 393–421.

Alterman, R. (1999). The closed room. *Mind, Culture, and Activity,* 8, 117–142.

Alterman, R., Zito-Wolf, R., & Carpenter, T. (1991). Interaction, comprehension, and instruction usage. *Journal of the Learning Sciences, 1* (4), 361–398.

Alterman, R., Zito-Wolf, R., & Carpenter, T. (1998). Pragmatic action. *Cognitive Science,* 22 (1), 53–105.

Bartlett, F. (1932). *Remembering: a study in experimental and social psychology*. Cambridge: Cambridge University Press.

Brooks, R. (1991). Intelligence without representation. *Artificial Intelligence,* 47, 139–159.

Carbonell, J. (1983). Derivational analogy and its role in problem solving. In *Proceedings of the third national conference on artificial intelligenc*e.

Chomsky, N. (1980). On cognitive structures and their development: a reply to piaget. In Piatelli-Palmarini, M., (Ed.) *Learning and Language* (pp. 35–52). Cambridge, MA: Harvard University Press.

Clancey, W. (1993). Situated action: a neuropsychological interpretation response to Vera & Simon. *Cognitive Science,* 17, 87–116.

Clark, H. H. (1996). *Using Language.* New York: Cambridge University Press.

Cognitive Science (1993). Situated action. *Cognitive Science,* 17 (1), 1–147.

Cole, M., & Engeström, Y. (1993). A cultural historic approach to distributed cognition. In Salomon, G., (Ed.) *Distributed Cognitions* (pp. 1–46). Cambridge, U.K.: Cambridge University Press.

Corkill, D. D. (1979). Hierarchical planning in a distributed environment. In *Proceedings of the sixth international joint conference on artificial intelligence* (pages 168–175).

Durfee, E., & Lesser, V. (1987). Using partial global plans to coordinate distributed problem solvers. In *Proceedings of the tenth international joint conference on artificial intelligence* (pp 875–883).

Fikes, R. E., Hart, P. E., & Nilsson, N. J. (1972). Learning and executing generalized robot plans. *Artificial Intelligence, 3,* 251–288.

Fikes, R. E., & Nilsson, N. J. (1971). A new approach to the application of theorem proving to problem solving. *Artificial Intelligence, 2* (3), 189–208.

Fisher, D. H. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine learning, 2,* 139–172.

Fodor, J. (1983). *The Modularity of the Mind.* MIT Press.

Garfinkel, H. (1967). *Studies in Ethnomethodology.* Polity.

Garland, A. (2000). *Learning to better coordinate in joint activities.* PhD thesis, Brandeis University.

Genesereth, M., Ginsberg, M., & Rosenschien, J. (1986). Cooperation without communication. In *Proceedings of the fifth national conference on artificial intelligence* (pp. 51–57).

Gentner, D. (1989). Mechanisms of analogical learning. In Vosniadou & Ortony, (Eds.) *Similarity and analogical reasoning* (pp. 199–241). London: Cambridge University Press.

Georgeff, M. P. (1983). The representation of events in multiagent domains. In *Proceedings of the third national conference on artificial intelligence* (pp. 125–129).

Gick, M. L., & Holyoak, K. J. (1980). Analogical problem solving. *Cognitive psychology, 12,* 306–355.

Gick, M. L. , & Holyoak, K. J. (1983). Schema induction and analogical transfer. *Cognitive Psychology, 15,* 1–38.

Gold, I., & Stoljar, D. (1999). A neuron doctrine in the philosophy of neuroscience. *Behavioral and brain sciences, 22,* 809–869.

Greenbaum, J., & Kyng, M., editors (1991). *Design at work.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Greeno, J. (1998). The situativity of knowing, learning, and research. *American Psychologist, 53,* 5–26.

Greeno, J., & Moore, J. (1993). Situativity and symbols: response to Vera & Simon. *Cognitive Science, 17,* 49–59.

Grosz, B., & Kraus, S. (1996). Collaborative plans for complex group action. *Artificial Intelligence, 86,* 269–357.

Grosz, B., & Kraus, S. (1999). The evolution of SharedPlans. In Rao, A., & Woolridge, M., (Eds.), *Foundations and theories of rational agency* (pp. 227–262). Kluwer.

Grosz, B., & Sidner, C. (1990). Plans for discourse. In Cohen, P. R., Morgan, J., & Pollack, M. E., (Eds.), *Intentions in communication* (pp. 417–444). Cambridge, MA: Bradford Books.

Hammond, K. J. (1990). Case-based planning: a framework for planning from experience. *Cognitive Science, 14,* 385–443.

Haynes, T., & Sen, S. (1998). Learning cases to resolve conflicts and improve group behavior. *International Journal of Human-Computer Studies, 48,* 31–49.

Huber, M. J., & Durfee, E. H. (1995). Deciding when to commit to action during observation-based coordination. In *Proceedings of the first international conference on multiagent systems* (pp. 163–170).

Huber, M. J., & Hadley, T. (1997). Multiple roles, multiple teams, dynamic environment: Autonomous Netrek agents. In Johnson, W. L., (Ed), *Proceedings of the first international conference on autonomous agents,* New York: ACM Press.

Hutchins, E. (1993). Learning to navigate. In Chaiklin, S., & Lave, J., (Eds.), *Understanding Practice* (pp. 35–63). Cambridge, U.K.: Cambridge University Press.

Hutchins, E. (1995a). *Cognition in the wild.* Cambridge, MA: MIT Press.

Hutchins, E. (1995b). How a cockpit remembers its speed. *Cognitive Science, 19,* 265–288.

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research, 4.*

Kambhampati, S., & Hendler, J. A. (1992). Control of refitting during plan reuse. *Artificial Intelligence, 55,* 193–258.

Knoblock, C. A. (1990). Learning abstraction hierarchies for problem solving. In *Proceedings of the eighth national conference on artificial intelligence* (pp. 923–928).

Kolodner, J. L. (1993). *Case-based reasoning.* San Mateo, CA: Morgan Kaufmann Publishers.

Kushmerick, N., Hanks, S., & Weld, D. (1995). An algorithm for probabilistic planning. *Artificial Intelligence, 76.*

Langley, P., & Allen, J. A. (1991). Learning, memory, and search in planning. In *Proceedings of the thirteenth annual conference of the cognitive science society* (pp. 364–369), Chicago, IL.

Lave, J. (1988). *Cognition and practice.* Cambridge: Cambridge University Press.

Lave, J., & Wegner, E. (1993). *Situated learning legitimate peripheral participation.* Cambridge, U.K.: Cambridge University Press.

Lesser, V., Fennell, R. D., Erman, L. D., & Reddy, D. R. (1975). Organization of the HEARSAY-II speech understanding system. *IEEE Transactions on Acoustics, Speech, Signal Processing,* 23:11–23.

Levesque, H. J., Cohen, P. R., & Nunes, J. H. T. (1990). On acting together. In *Proceedings of the eighth national conference on artificial intelligence* (pp. 94–99).

Lewis, D. K. (1969). *Convention: A philosophical study.* Harvard University Press.

Mataric, M. (1992). Designing emergent behaviors: from local interactions to collective intelligence. In *From Animals to Animats 2* (pp. 432–441).

McCarthy, J. (1958). Programs with commonsense. In *Proceedings of the symposium on mechanization of thought processes* (pp. 77–84).

McCarthy, J. (1968). Programs with common sense. In Minsky, M., (Ed.), *Semantic information processing* (pp. 403–418). MIT Press.

McCarthy, J. (1977). Epistemological problems in artificial intelligence. In *Proceedings of the fifth international joint conference on artificial intelligence.*

Mead, G. H., *Mind, self, and society.* Chicago: University of Chicago Press.

Minsky, M. (1975). A framework for representing knowledge. In Winston, P., (Ed.), *The psychology of computer vision* (pp. 211–277). New York: McGraw-Hill.

Minton, S., Carbonell, J. G., Knoblock, C. A., Kuokka, D. R., Etzioni, O., & Gil, Y. (1989). Explanation-based learning: a problem-solving perspective. *Artificial Intelligence, 40,* 63–118.

NagendraPrasad, M. V., Lesser, V. R., & Lander, S. (1995). Retrieval and reasoning in distributed case bases. Technical Report CS TR 95–27, University of Massachusetts.

Newell, A., & Simon, H. (1972). *Human problem-solving.* Englewood Cliffs, NJ: Prentice-Hall.

Norman, D. A. (1988). *The psychology of everyday thing*s. Basic Books.

Ohko, T., Hiraki, K., & Anzai, Y. (1996). Learning to reduce communication costs on task negotiation among multiple autonomous mobile robots. In Weiβ, G., & Sen, S., (Eds.), *Adaptation and learning in multi–agent systems,* Lecture Notes in Artificial Intelligence (pp. 177–190). Berlin: Springer Verlag.

Okada, T., & Simon, H. (1997). Collaborative discovery in a scientific domain. *Cognitive Science, 21* (2), 109–146.

Quillian, R. (1968). Semantic memory. In Minsky, M., (Ed.), *Semantic information processing* (pp. 227–270). Cambridge, MA: MIT Press.

Roschelle, J. (1992). Learning by collaborating: convergent conceptual change. *Journal of the Learning Science, 2* (3), 235–276.

Rumelhart, D., & McClelland, J. (1986). *Parallel distributed processing: explorations in the microstructures of cognition (Vol. 1)*. Cambridge, MA: MIT Press.

Sacerdoti, E. (1974). Planning in a hierarchy of abstraction spaces. *Artificial Intelligence, 5,* 115–135.

Sacks, H., Schegloff, E., & Jefferson, G. (1974). A simplest systematics for the organization of turn-taking in conversation. *Language, 50,* 696–735.

Schank, R. (1982). *Dynamic memory: a theory of reminding and learning in computers and people.* Cambridge: Cambridge University Press.

Schank, R., & Abelson, R. (1977). *Scripts, plans, goals, and understanding.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Schuler, D., & Namioka, A., editors (1993). *Participatory design.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Sen, S., & Sekaran, M. (1998). Individual learning of coordination knowledge. *Journal of Experimental and Theoretical Artificial Intelligence, 10.*

Sen, S., Sekaran, M., & Hale, J. (1994). Learning to coordinate without sharing information. In *Proceedings of the twelfth national conference on artificial intelligence* (pp. 426–431).

Stone, P., & Veloso, M. (1998). Towards collaborative and adversarial learning: a case study in robotic soccer. *International Journal of Human-Computer Studies, 48,* 83–104.

Suchman, L. A. (1987). *Plans and situated actions.* Cambridge: Cambridge University Press.

Sugawara, T. (1995). Reusing past plans in distributed planning. In *Proceedings of the first international conference on multiagent systems* (pp. 360–367).

Tambe, M. (1997). Towards flexible teamwork. *Journal of Artificial Intelligence Research, 7,* 83–124.

Tomasello, M., Kruger, A., & Ratner, H. (1993). Cultural Learning. *Behavioral and Brain Sciences*, 16, 495–552.

VanLehn, K. (1989). Problem solving and cognitive skill acquisition. In Posner, M., (Ed.), *Foundations of cognitive science* (pp. 527–580). Cambridge, MA: Bradford Books, MIT Press.

Veloso, M., & Carbonell, J. (1993). Derivational analogy in PRODIGY: automating case acquisition, storage, and utilization. *Machine Learning* (pp. 249–278).

Vygotsky, L. S. (1978). *Mind in Society*. Cambridge, MA: Harvard University Press.

Vygotsky, L. S. (1986). *Thought and language.* Cambridge, MA: Cambridge University Press. Russian edition published in 1934.

Wilkens, D., & Myers, K. (1998). A multiagent planning architecture. In *Proceedings of the fourth international conference on artificial intelligence planning systems,* pages 154–162.

Winograd, T. (1996). Introduction. In Winograd, T., (Ed.), *Bringing design to software* (pp. xiii–xxv). Addison-Wesley.

Yanco, H., & Stein, L. A. (1993). An adaptive communication protocol for cooperating mobile robots. In *Animals to Animats: proceedings of the second international conference on the simulation of adaptive behavior* (pp. 478–485). MIT Press.