UCI-ITS-AS-WP-02-18

An Agent-Based Activity Microsimulation Kernel Using a Negotiation Metaphor

UCI-ITS-AS-WP-02-18

Craig R. Rindt
James E. Marca
Michael G. McNally

Department of Civil Engineering and
Institute of Transportation Studies
University of California, Irvine
crindt@uci.edu, jmarca@uci.edu, mmcnally@uci.edu

August 2002

Institute of Transportation Studies
University of California, Irvine
Irvine, CA 92697-3600, U.S.A.
http://www.its.uci.edu

An Agent-based Activity Microsimulation Kernel Using a Negotiation Metaphor

Craig R. Rindt, James E. Marca, and Michael G. McNally

Institute of Transportation Studies
University of California, Irvine
Irvine, CA 92697
tel: (949) 824-6571
fax: (949) 824-8385
crindt@uci.edu

August 1, 2002

Abstract. This paper describes the development and implementation of an agent-based activity microsimulation kernel based upon the concept that human activity is the negotiated interaction of socially and physically situated individuals and organizations. The kernel uses a modification of the contract-net protocol from the distributed artificial intelligence literature to represent the "physics" of interaction in human activity settings. The details of the kernel design and implementation are discussed.

Keywords: Agent-based micro-simulation, Interpersonal interaction, constraints

Wordcount: $5300 + (2 \text{ figures}) \cdot (250 \text{ words/figure}) = 5800$

INTRODUCTION

Activity-based travel demand modeling

The evolution of travel demand modeling has progressed from broad spatial interaction models, to finer grained trip distribution and traffic assignment models of the four-step transportation planning process, and to the now entrenched activity-based models at the core of the next generation of transportation forecasting models. This evolution has been driven by a need for greater sensitivity to policies that affect more than just the broad characteristics of urban form, and target the mechanisms that produce human travel behavior.

Where conventional four-step models express the demand for travel directly as a function of demographics and the organization of land-uses, activity-based models add another layer, expressing the need to travel as a function of the need to perform activities. These activities are themselves a function of social roles and norms and the distribution of environmental resources among various land-uses and social organizations. The result is that while the conventional process captures the effects of some of these constraints on travel implicitly (e.g., the trip distribution step considers the constraint of time on trip decisions), the activity-based approach treats these constraints explicitly.

The activity-based modeling literature, however, has been dogged by problems of its own. The reductionist trend of adding model detail simultaneously adds complexity. Complexity isn't necessarily something to be shunned as long as it can contribute descriptive or explanatory power to the model, but the results of complexity science suggest that such advantages are likely to be related to emergent or self-organizing behaviors in the system whose responses to policy cannot be captured without a holistic model of the system. To be effective, however, such a holistic model must indeed be "whole" with respect to those dynamics. While many advances have been made in understanding pieces of the travel behavior puzzle, other pieces remain illusive. The result is that effective operational models of the human activity that leads to travel are limited at best, and are likely to remain that way until solutions for these missing pieces have been found.

The Missing Pieces

The effects of inter-personal, inter-environmental, and inter-activity linkages have proven difficult to capture in existing models. Most proposed microsimulation models have significantly simplified the relationships between agents in the system, generally reducing the problem to the aggregation of scheduling decisions made by independent individuals or households. The scheduling problem is solved using either analytical *(1, 2)*, econometric *(3, 4)*, or computational *(5, 6, 7)* models of how individuals or households schedule their activities over the course of a day given fixed inputs that include the activities to be performed and the state of the environment in which they can be performed. As a result of these limitations, the difference between what current models are capable of and what they need to do is similar to the difference between all-or-nothing and user equilibrium solutions to the traffic assignment problem. Linkages between the choices made by multiple individuals are either not represented or are taken as fixed inputs to the model. Examples of this abound. For instance, the household activity pattern problem (HAPP) *(1)*, in which constraints are used exclusively to define the problem. Kitamura et al. *(4)* use a Markovian approach to model the transition between activities, but necessarily omit linkages. Bowman and Ben-Akiva *(3)* use a deeply nested discrete choice formulation that is difficult to adapt to dynamic linkages.

The A Learning-BAsed TRansportation Oriented Simulation System (ALBATROSS) computational process model (CPM) *(7)* uses a predefined activity agenda which is then scheduled within an environment that does not include interpersonal linkages, though ongoing work is promising Joh et al. *(8)*.

Ultimately, "soft constraints" on activity, involving linkages between agents and their environment, have proven difficult to capture in activity scheduling models. These limitations reduce these models' usefulness for the analysis of policy measures whose responses may be affected by such linkages. Such modeling rigidity ultimately omits portions of the potential activity pattern space which may actually be feasible with slight modifications of the constraint space.

One Possible Solution

To solve the problem of modeling linkages, we've developed a conceptual model of human activity systems using the agent-based modeling paradigm. The details of this model are described in a companion paper *(9)*. This model focuses on the notion that human behavior is *adaptive*, and that the engine of adaptation is the individual's assessment of his or her imbalance with the environment — what Fried et al. termed person-environment (p-e) fit. The p-e fit can be viewed as the difference between what an individual's environment demands of them, and what an individual is capable of in that environment. Individuals are assumed to utilize experiential learning to formulate internal models of the environment, with an emphasis on satisfying personal and role-driven needs by interacting with the environment.

This paper describes the development of a simulation kernel for representing this interaction as a negotiation between autonomous agents. The kernel uses a modification of the contract-net protocol from the distributed artificial intelligence literature to represent the "physics" of interaction in human activity settings. We demonstrate how this kernel can capture many of the complex interactions in urban settings that are difficult to represent in existing modeling approaches, including coupling, authority, and other "soft" constraints that are difficult to represent analytically.

AN ACTIVITY MICROSIMULATION KERNEL

Background

In their seminal article, Davis and Smith *(11)* introduced the negotiation metaphor for distributed problem solving. This took the (even then) familiar metaphor of distributed problem solving as a group of human experts working together one step further by describing the process these experts use to interact to solve the problem at hand. They argue that this process is best viewed as a negotiation in which the group of experts cooperate with each other in order to solve the problem. They focus on the interaction between agents, the distribution of tasks, and the integration of results, defining distributed problem solving as:

> ...a cooperative activity of a group of decentralized and loosely coupled knowledge sources. They cooperate in the sense that no one of them has sufficient information to solve the entire problem.

At the center of their framework is a negotiation protocol in which agents can: share information, evaluate that information from their local perspectives, and jointly come to an agreement

about the task at hand. This framework has developed into the contract net protocol (CNP), which has seen wide application in distributed computation. In CNP, tasks (or contracts) to be performed are announced by a contract manager to contractors who bid to perform the task. The manager awards the contract to the contractor with the "best" bid according to some criteria. Here

> task distribution is viewed as an interactive process, a discussion carried on between a node with a task to be executed and a group of nodes that may be able to execute the task. We describe the kinds of information that must be passed between nodes during the discussion in order to obtain effective problem-solving behavior. This discussion is the origin of the negotiation metaphor: Task distribution is viewed as a form of contract negotiation *(11, p 1)*

The original specification had no formal model for the announcing, bidding, and awarding decisions, all of which are likely to be specific to a particular problem domain. For instance, Sandholm *(12)* formalized the negotiation process based on local marginal cost calculations in solving a heterogeneous fleet multi-depot routing problem in which a set of delivery companies compete for the delivery business of a set of geographically dispersed factories.

In the end, the point of the CNP is to define a protocol for information sharing that facilitates the distributed solution of a task. The real contribution of this work, however, is the introduction of the negotiation metaphor. The motivating factor for cooperation is that for one agent to achieve its goals, it requires the cooperation of other agents. To guarantee this cooperation, negotiation is used to establish the contribution of each agent to the larger problem. Only through such negotiated agreements can the manager be confident that it will complete its task.

Activity Participation as Negotiation

The negotiation metaphor is certainly intriguing for activity systems analysis. As noted above, Davis and Smith *(11)* argued that the human process of distributed problem solving involves negotiation, and therefore if we want to automate distributed problem solving we should structure it as a negotiation. In the case of modeling human activity, we'll turn this argument on its head – instead of artificial intelligence (AI) being inspired by social processes, we'll model social processes using the techniques of AI. Recalling our definition of activity from above, the human condition is characterized by a series of negotiations with the other agents in the environment to engage in certain types of interaction (activities). The result of these negotiations, from a particular agent's perspective, is a time-series of activities that define the agent's behavior as interactions with other agents in the environment. Thus, human existence is negotiation. As a result, AI techniques that model negotiation should provide a useful starting point for representing human existence.

Interpreting activity engagement from a distributed problem solving perspective is interesting. In this context, activity engagement is the process used to solve the problem of activity completion, which in turn serves other economic and social ends for the participating agents. The process involves the distribution of engagement tasks among a set of agents, with the global goal being the completion of the activity. In the engagement "tasks," each agent must commit resources under its control for some non-zero time period to the completion of the activity. For instance, a simple work activity may involve three agents: an employee, an employer, and a workplace. The employee controls some abstract work-related processing resource, the knowledge and energy to perform certain work-related tasks for the employer. The employer controls a set of physical and

social resources necessary for the employee's work task and compensates the employee for the work. These resources are embodied by the workplace provided by the employer which serves as the setting, or locale, for the work activity *(13)*. For the work activity to happen, both the employee, employer, and implicitly, the workplace, must be available and willing to engage in the task, and herein lies the negotiation.

Looking at the problem as a contract-net permits the consideration of interesting questions related to activity motivation that will ultimately impact on activity generation. Recall that in the CNP, the manager is seeking to solve a particular problem by coordinating the solution of sub-tasks. Here, the sub-tasks are the individual contributions that agents make to the performance of the activity.

Adapting this concept to the simulation of activity participation is not difficult. Each of the potential activity participants is a potential contractor for the performance of the activity. The contract manager is an abstraction that represents the state of the negotiation between agents based on the physical and logical conditions that determine whether an activity can occur. These conditions are centered around resource accessibility. For instance, a particular person's work activity might require that they have access to a large machine that is owned by their employer and housed at their workplace. To engage in this activity, therefore, the individual must travel to his workplace (spatial constraint) and obtain permission to access the machine from his employer (authority constraint). Once all of these conditions are met (via negotiation), the activity will occur and continue until the conditions defining the interaction no longer support the activity – e.g., the employee choose to stop, or the employer closes down the factory. If the activity also required another agent, a co-operator for the machine, we could easily introduce a coupling constraint by adding her to the participation criteria.

In this model, therefore, the human-agent's experience is characterized by the contemplation of what "contracts" to "bid" on in order to meet some set of goals derived from the individual's social environment. These potential contracts can either by self-generated (I want to eat something), or derive from role commitments (I have to pick my child up from football practice). An agent's success in achieving its goals depends on its ability to anticipate the opportunities presented by its environment. An agent representing a person would therefore need: the ability to perceive information about environmental opportunity, a model to anticipate such opportunity in the future, and a decision process that selects behavior based upon the anticipatory model of the environment to achieve some set of goals derived from a motivational core.

Simulation Kernel Design

The simulation kernel consists of an implementation of the CNP tailored to the negotiation of activity participation between adaptive agents. At the level of the kernel, all agents are treated identically – that is, there is no distinction between person-agents, social-group agents (like an employer), or land-use agents. To make this clear, the kernel refers to all agents as `ActivityPrincipals`, or simply principals, indicating that all agents have the potential to be principal components of the interaction that defines a particular activity.

To apply the CNP to the activity systems modeling domain, we will need to specify the details of the negotiation process, including several extensions to the original CNP specification as follows:

- *negotiation criteria:* Several authors have noted that the CNP is more a protocol for coordination than for negotiation as it does not define mechanisms for conflict resolution *(14, 15)*. Additional layers are necessary to provide a framework for negotiation. In the activity modeling context, the simplest form of negotiation involves just the concurrent agreement to a set of predefined criteria specifying the terms of the activity to be performed. Extensions to this basic form of negotiation would additionally allow agents to resolve conflicts about those terms.

- *contracts involving multiple agents:* In this domain, a contract is really an agreement between at least two principals (e.g., a person and a land-use, and possibly other agents). The agreement involves a commitment of resources controlled by each principal: e.g., time, labor, money in the case of an person; goods, services, and other resources in the case of a locale.

- *non-binding contracts:* Agents in activity systems tend to have relative autonomy in their activity participation, including the ability to terminate some activities under way at will (e.g., a casual social engagement). In conventional CNP, a contractor cannot reject a job once a contract has been negotiated. This restriction is necessary since tasks are generally part of the solution to larger problem. In the application here, however, the system is a model and not intended as an optimizing problem solver. As such the completion of tasks (activities) need not be guaranteed in order to meet some global goal (though such a requirement may be an interesting application of the model). If an agent is unable to perform a required activity, that agent will need to better adapt to its environment.

- *binding contracts:* This is not to imply that there will not be situations in which contracts are binding. Such guarantees can be used to represent a variety of situations. Consider a travel activity using a train. In this case, the principal effectively surrenders control to the activity (really the locale here) until it terminates, or the person is given the opportunity to terminate it, e.g., by getting off the train when it reaches the next stop. This is a physical constraint (can't get off a moving train), but one can also envision social constraints also making activities binding.

- *simultaneous activities:* In some cases, it may be possible for agents to perform multiple activities simultaneously. By our definition of activity, recall that two principles are jointly applying resources under their control to the achievement of individual goals. Under this definition, if that application does not tie up other resources at the agent's disposal, the agent should be able to apply them to another activity as long as other criteria can be met.

These modifications to the CNP are not difficult to achieve, primarily because the protocol is no longer being used to solve a particular distributed problem as much as its serving as the mechanism for interaction between autonomous agents.

Simulation Kernel Implementation

We have implemented the activity kernel CNP as a Concurrent, Hierarchical, finite State Machine (CHSM) *(16)* to ensure functional consistency and prevent locking. In its present form, the kernel

is built atop the Swarm agent-based simulation toolkit *(17)*, which provides a convenient event-driven simulation kernel, useful analysis and display capabilities, and is open source so is easily extensible to meet evolving research needs.

The state machine that governs negotiation is shown in figure 1. The machine consists of a single state set containing $N + 1$ concurrent state clusters (or sub-state machines), one for an abstract manager object, the `AgentActivity`, which stores the global state of the negotiation (i.e., whether the activity is viable, initiated, etc.) and one each for the N potential principals that are negotiating interaction. The top-level state set is a dynamic CHSM whose size is controlled by the `AgentActivity` cluster according to `addPrincipal` events it receives during its `proto` state. An externally directive `activityRequest` event, generally sent by the agent who constructed the Activity CNP, causes the `AgentActivity` to send an `activityAnnouncement` to all `ActivityPrincipals` in the CHSM for consideration – we will consider how they respond to these announcements in a moment. In the mean time, the `AgentActivity` transitions to its `requested` state and waits for `offer` or `removePrincipal` events from the principals it has announced to.

[FIGURE 1 about here.]

Each event it receives may impact one of two sets of relevant negotiation criteria that were specified in the creation of the activity. The first is a set of *viability criteria* that indicate whether the activity can ever be successfully negotiated. Generally, the *viability criteria* are a set of conditions on the principals that remain in the CHSM. For instance, a person's work activity may require that the his workplace is involved. If the *viability criteria* are *not* met, the `AgentActivity` cluster transitions to the `balk` state, which causes all remaining principals to be notified that the negotiation is invalid, ultimately leading to the destruction of this activity without it ever being completed.

The second set of negotiation criteria are the *initiation criteria* that indicate whether the activity can start given the current set of principals who have offered to participate (as opposed to the set who have been notified). If these criteria are met, the `AgentActivity` cluster transitions to the `underway` state, which notifies the principals who have offered to they are starting the activity. The `AgentActivity` will remain in this state until either the terms of the activity contract are completed, triggering a `finishEvent`, or until external events cause a third set of *continue criteria* (also specified during creation) to no longer be met) In the latter case, the `AgentActivity` transitions to the `breech` state, notifying the acting principals of the early termination of the activity. The former case corresponds to a successful completion of the activity, according to the terms specified, and the `AgentActivity` transitions to the complete state, subsequently notifying all acting principals of successful completion.

The `ActivityPrincipals` involved in this negotiation also interact according to a state chart. The principal starts a negotiation in an `idle` state, waiting for events. When a principal receives an `activityAnnouncement` event, it transitions to the `considering` state and fires either a `canPerform` or `canNotPerform` event to itself depending on what the agent's underlying behavioral model dictates. If it cannot (or will not) perform the activity, the principal transitions to the `complete` state and sends a `removePrincipal` event to the `AgentActivity` to remove itself from the principals set, thus terminating this principal's involvement with the negotiation. If it will perform the activity, the principal instead transitions to the `have_offered` state and sends an `offer` event to the `AgentActivity` cluster. Both of these events may cause

state transitions in the `AgentActivity` as discussed previously. Once in the `have_offered` state, the principal waits for state transition events from the `AgentActivity` indicating the activity is underway or it has balked. Additionally, because the principal may be involved in other negotiations that are "competing" for the agents participation, the agent may determine it can not perform the activity even after committing to it. In this case, it again transitions to the `complete` state and removes itself from the principals set. If the `AgentActivity` moves into the `underway` state, and this principal has not committed elsewhere, the principal transitions to the `committed` state, locking any resources under its control that must be applied to the activity.

We can now see how the kernel provides each of the required extensions to the CNP discussed in the previous section though the addition of a negotiation criteria during the announcement and bidding stages of the protocol. The use of generalized *viability and initiation criteria* to govern contract awarding (activity participation) makes it easy for the manager to allow (even require) multiple agents to participate. Unless the collection of active offers meets these criteria, the activity cannot begin. Similarly the use of generalized *continue criteria* easily permit the modeling of non-binding contracts such that agent autonomy can be preserved. Modeling of *simultaneous activities* is also supported through the specification of criteria. For instance, the criteria for a childcare activity might require only that the principal is co-located with the children, but not require any direct attention otherwise. This attention (a type of resource) can be applied to other activities, such as a work activity or household maintenance.

To summarize, the activity kernel is structured as a CHSM that tracks the negotiation of interaction commitment meeting some criteria between a collection of principals. The kernel requires that the following data are given as inputs:

- the *list of principals* to announce the activity to,
- the *viability criteria* for the activity,
- the *initiation criteria* for the activity,
- the *continue criteria* for the activity.
- the interactive offer or removal events describing how the principals respond to the negotiation.

In a particular simulation application, these inputs would be generated by a collection of self-directed, interacting, agents representing people, places, and social institutions in the urban environment. The kernel expects criteria to be defined as tests (actually function objects passed to the kernel during the creation of the activity manager) that can be applied on collections of principals.

This approach allows for maximum flexibility and leaves all implementation details to the user of the kernel. At the highest level, the behavior produced by the kernel depends on the activity type and the agent responses, just as one would expect. By defining activities as interactions governed by particular criteria, however, the kernel provides a greater level of expressiveness than existing discrete event approaches.

Creating Activities

The activity simulation kernel makes no explicit distinction between the agent types of the principals involved in an activity. Instead, activities are distinguished by the list of principals considering

the activity and by the criteria that governs the negotiation for participation. The list of principals involved will depend on the particular activity being negotiated and on what agent initiated the negotiation. Figure 2 shows a typical activity negotiation, execution, and completion time line involving a person-agent and a locale-agent, the latter of which provides the resources for a particular task.

[FIGURE 2 about here.]

In this case the `Person`-agent initiates the negotiation by creating the `AgentActivity` object, adding itself and the `Locale` agent as potential principals, setting the criteria that govern the negotiation, and sending the `requestActivity` message to the `AgentActivity`. At this point the `AgentActivity` becomes an autonomous agent that announces the activity for bid to the principals and controls their negotiation. In this example, assume the initiate criteria are simply that both the `Person`-agent and `Locale`-agent offer to perform the activity, and the continue criteria perform the same test on the acting participants. Once both have returned an offer, the `AgentActivity` starts the activity, and continues until the terms of the activity are met, after which the finish message is sent to the acting principals. They both receive this message (the order is not important) and remove themselves from the activity, causing it to terminate and destroy itself.

Principal Agent Specialization

It should be clear to the reader that the simulation kernel *does not* implement a behavioral model along the lines of ALBATROSS *(7)*. Instead, it provides a framework for placing such models in a dynamic, interactive setting. The principals that the kernel operates on are abstractions whose responses to the negotiation process are governed by their specialization. Those readers familiar with object-oriented programming will recognize this specialization as sub-classing. To use the kernel meaningfully, the analyst must define a set of specialized agent types representing the entities of interest for a particular model application. The specialized principals must implement methods for deciding whether or not to bid on activity participation. This may involve both local considerations, such as other tradeoffs with other activities available to the agent, as well as strategic ones, such as linkages between activities – e.g., one must travel to work before he can work.

In our development of the kernel, we have focused on two main principal specializations – a person-agent and a locale-agent. Person agents attempt to negotiate activities with other agents in order to obtain some payoff. Locale agents are relatively passive agents that maintain a set of temporally restricted resources necessary for various activity types (this effectively defines time-windows for activity). This sets a framework for developing a dynamic, interactive simulation using a population of agents who seek to learn how to maximize their payoff from the environment in a manner similar to that reported by Joh et al. *(8)*.

CONCLUSION AND FUTURE WORK

This paper reports on the development of a simulation kernel for agent-based activity micro-simulation based on the re-characterization of human activity as interaction between autonomous

entities. The kernel provides a structured model of agent interaction that as the result of negotiation between agents, and is capable of representing a broad class of the interpersonal and person-environment interactions that characterize activity systems.

In its present form, the kernel places limited restrictions on the negotiation that occurs between agents. This is both an advantage and a disadvantage, maximizing the expressive range of the kernel, but placing a greater burden on users developing models on top of it. We are considering future work to layer common negotiation protocols on top of the kernel in order to better represent the processes by which agents cooperate to define activities involving coupling constraints.

Finally, the kernel is *not* a model of behavior, nor is it useful for analysis, understanding, or prediction of urban dynamics by itself. It does, however, provide crucial functionality for the development, testing, and application of such models, which we are developing in concurrent research.

REFERENCES

[1] Wilfred W. Recker. The household activity pattern problem: General formulation and solution. *Transportation Research, Part B: Methodological*, 29B(1):61–77, 1995.

[2] Wilfred W. Recker. A bridge between travel demand modeling and activity-based travel analysis. *Transportation Research, Part B: Methodological*, (forthcoming), 2000.

[3] John L. Bowman and Moshe Ben-Akiva. Activity-based disaggregate travel demand model system with activity schedules. *Transportation Research, Part A: Policy and Practice*, 35A (1):1–28, Jan 2001.

[4] Ryuichi Kitamura, Cynthia Chen, R. M. Pendyala, and R. Narayanan. Microsimulation of daily activity-travel patterns for travel demand forecasting. *Transportation*, 27:25–51, 2000.

[5] Wilfred W. Recker, Gary S. Root, and Michael G. McNally. A model of complex travel behavior: Part I–theoretical development. *Transportation Research*, 20A(4):307–318, 1986.

[6] Dick Ettema, A. Borgers, and Harry Timmermans. Simulation model of activity scheduling behavior. *Transportation Research Record*, 1413:1–11, 1993.

[7] Theo Arentze and Harry Timmermans. *ALBATROSS: a learning based transportation oriented simulation system.* European Institute of Retailing and Services Studies, Eindhoven, 2000.

[8] Chang-Hyeon Joh, Theo Arentze, and Harry Timmermans. Modeling individuals' activity-travel rescheduling heuristics: Theory and numerical experiments. In TRB TRB *(18)*.

[9] Craig R. Rindt, James E. Marca, and Michael G. McNally. Toward dynamic, longitudinal, agent-based microsimulation models of human activity in urban settings. Working Paper, July 2002.

[10] M. Fried, J. Havens, and M. Thall. Travel behavior: A synthesised theory. Technical report, The National Cooperative Highway Research Program, Washington, DC, 1977.

[11] Randall Davis and Reid G. Smith. Negotiation as a metaphor for distributed problem solving. *Artificial Intelligence*, 20:63–109, 1983.

[12] Tuomas Sandholm. An implementation of the contract net protocol based on marginal cost calculations. In *Proceedings of the Eleventh National Conference on Artificial Intelligence, Washington, DC, USA, 11-15 July 1993*, pages 256–262. AAAI Press, July 1993. URL `http://icities.csd.uoc.gr/related/papers/implementation.pdf`.

[13] Anthony Giddens. *The Constitution of Society*. Polity Press, Cambridge, 1984.

[14] N. Jennings. Specification and implementation of a belief desire joint-intention architecture for collaborative problem solving. *Journal of Intelligent and Cooperative Information Systems*, 2(3):289–318, 1993.

[15] Shaw Green and Fergal Sommers. Software agents: A review. Technical report, Intelligent Agents Group, Computer Science Department, Trinity College, Dublin, 1997. URL `http://www.cs.tcd.ie/research_groups/aig/iag/toplevel2.html`.

[16] Paul J. Lucas. An object-oriented language system for implementing concurrent, hierarchical, finite state machines. Technical Report UIUCDCS-R-94-1868, Department of Computer Science, University of Illinois, Urbana-Champaign, August 1994.

[17] N. Minar, R. Burkhart, C. Langton, and M. Askenazi. The Swarm simulation system, a toolkit for building multi-agent simulations, June 1996. URL `http://citeseer.nj.nec.com/minar96swarm.html`.

[18] TRB. *Proceedings of the* 81^{st} *annual meeting of the Transportation Research Board*, Washington, D. C., 2002. TRB.

LIST OF FIGURES

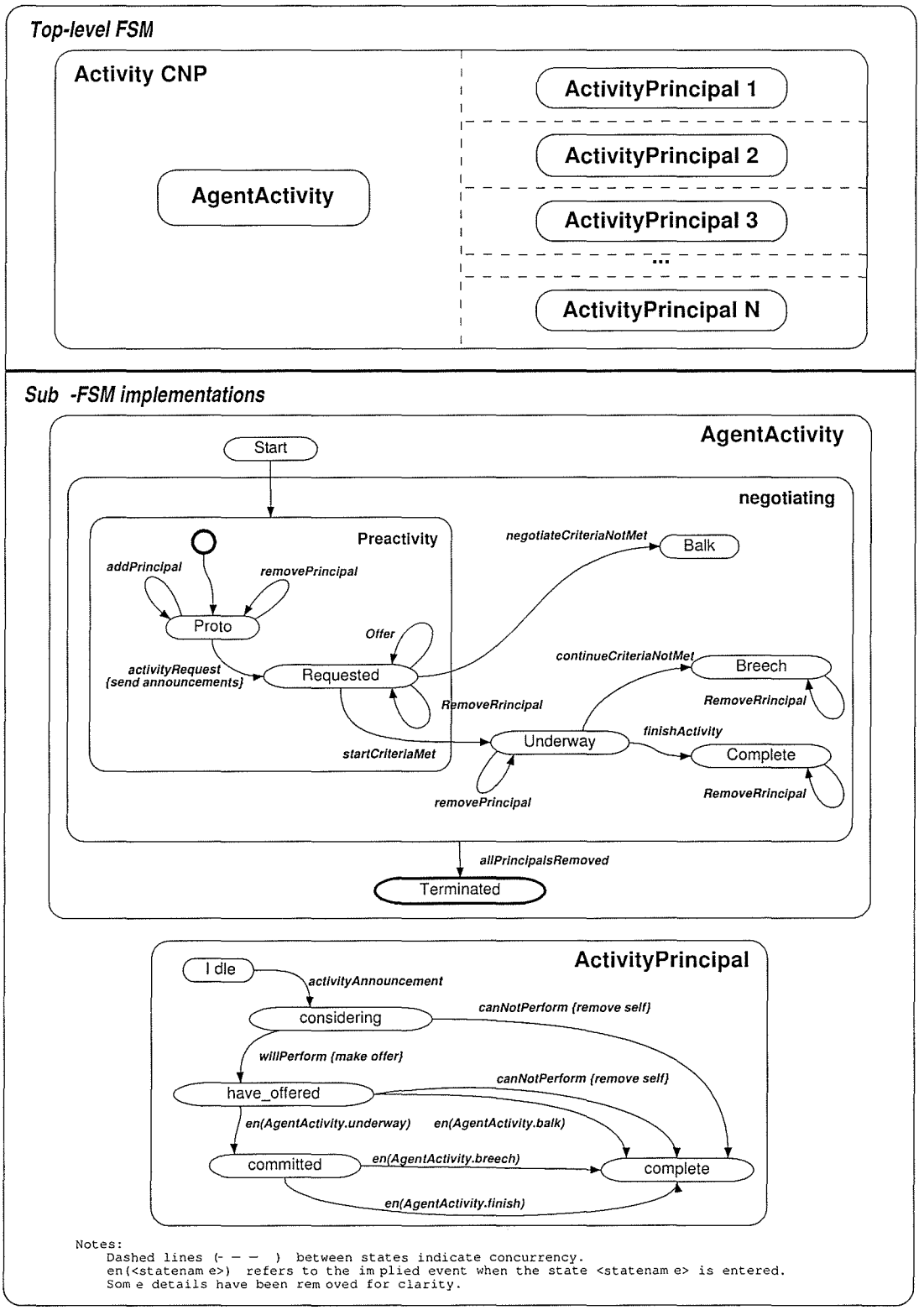

FIGURE 1: Activity negotiation as a concurrent, hierarchical, finite state machine

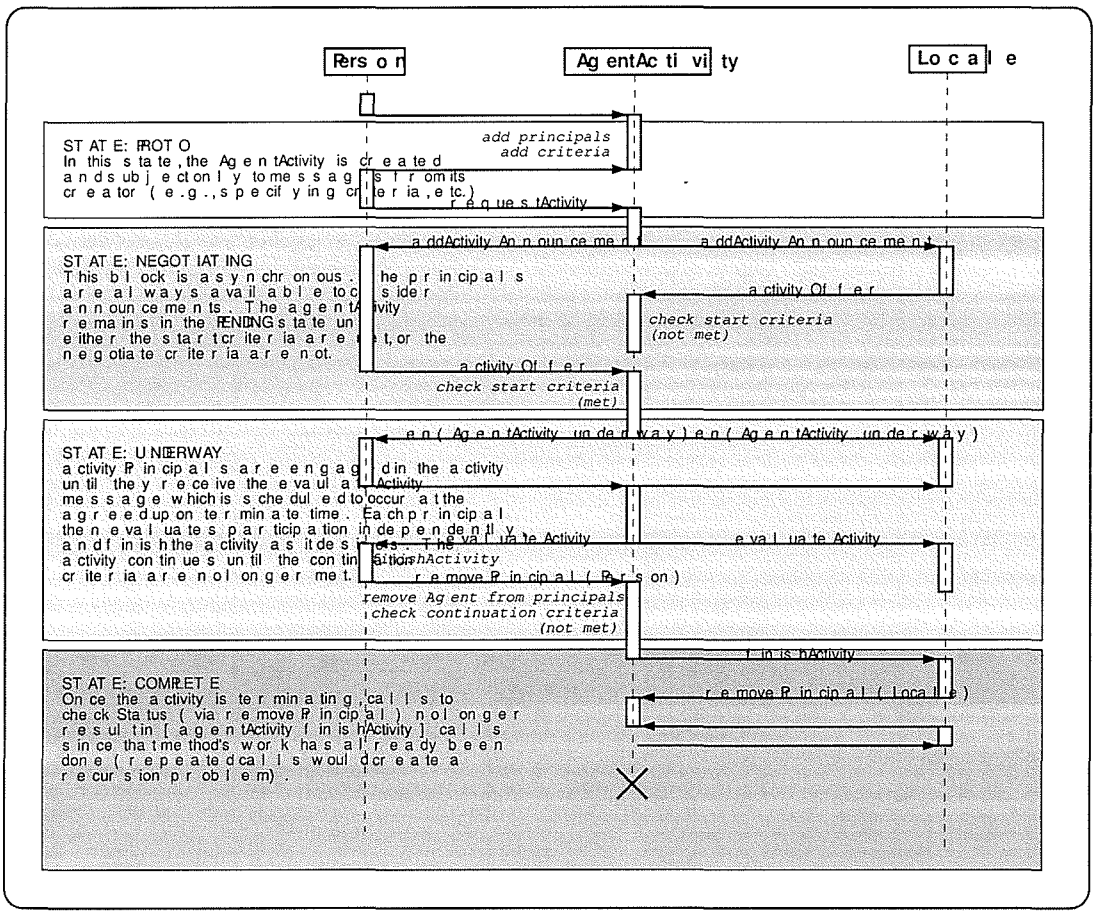

FIGURE 2: Typical activity negotiation process