IWMAS98 Position Paper: Deconstructing Negotiation

Van Parunak, CEC/ERIM
vparunak@erim.org

Abstract

The workshop proposes to discuss the opportunities and pitfalls of negotiation, and appears to consider negotiation as a subset of agent interaction. In my experience in the DAI/MAS community, “negotiation” is thought of as a special capability of particularly sophisticated agents. In the spirit of provoking thought and discussion, this white paper argues that the definition of negotiation proposed in the CFP (“interactive communication for arranging activity”) in fact can be applied to any interaction of an agent with its environment, even those that are not usually considered negotiation. This essentially deconstructive claim is based on three Provocative Claims (PC’s), for which I shall (in the interests of a spirited workshop) try to argue:

1. All communication is mediated, and thus any agent action is potentially communicative.
2. All agent activity (and thus all communication) is non-deterministic.
3. All agent activity is in fact interactive communication.

Against this background, I suggest the importance of explicitly including intentionality in the definition of negotiation, though even this more focused definition still has problematic implications. Finally I discuss the options and pitfalls that point the way to further research.

1. Provocative Claims

Merriam-Webster’s definitions of “communication” most closely related to the MAS sense of the word are “1 : an act or instance of transmitting” and “3 a : a process by which information is exchanged between individuals through a common system of symbols, signs, or behavior <the function of pheromones in insect communication>; also : exchange of information”[11]. The Provocative Claims in this section try to make explicit that interactive communication among agents is much broader than students of negotiation in agent-based systems commonly consider.

1.1. All Communication is Mediated

Agents cannot be considered independently of the environment in which they exist and through which they interact [4, 13]. This insight is often ignored in work on computational multi-agent systems, where researchers consider the environment as a passive communications framework and everything of interest is relegated to one or another of the agents. Even such purely electronic environments often prove embarrassingly active when an application is scaled up for real-world use: networks go down, connections drop out, messages are delayed for nondeterministic lengths of time. In addition, engineers designing agents for non-electronic domains (such as factory automation) must consider the environment explicitly. In the YAMS system [14], our analysis of the protocol between agents originally ignored part movement between two workstations, with the result that a protocol “proven” not to deadlock actually deadlocked because information passed between agents through the physical environment. Thus

I argue that all communication among agents is mediated through the environment, often through multiple channels. In full generality, we should define a Multi-Agent System as a three-tuple: a set of Agents, an Environment, and a Coupling between them:

$$MAS = \langle Agents, Environment, Coupling \rangle$$

Let's consider each of these in more detail.

1.1.1. Agents

Each agent is a four-tuple:

$$Agents = \{Agent_1, \ldots, Agent_n\}$$

$$Agent_i = \langle State_i, Input_i, Output_i, Process_i \rangle$$

- $State_i$ is a set of values that completely define the agent. The structure, domains, and variability of these values are not constrained by this definition, and differences in these features are responsible for much of the interesting variation among different kinds of agents.
- $Input_i$ and $Output_i$ are subsets of State, whose variables are coupled (in a way to be defined more specifically below) to the environment. We will occasionally speak of an agent's sensors and effectors. These are mechanisms that implement the coupling between the environment and the agent's Input and Output variables, respectively.
- $Process_i$ is an autonomously executing mapping with the *ansatz* $State(t+1) = Process(State(t))$ (for a Discrete Event agent) or $\partial State/\partial t = Process(State(t))$ (for a Time-Based agent) that changes the agent's State. "Autonomously executing" means that the Process runs without being invoked from any outside entity. In computational terms, an agent has its own virtual CPU.

The existence of Input and Output imply that an agent is a bounded process, and Input and Output cross this boundary. Input and Output relate the agent immediately to the environment, and only mediately (through the environment) to other agents. In the most general sense, an agent's communication is the trajectory of its Input and Output. Messages as usually understood are simply subsets of Input and Output that are coupled to some subset of the Environment that happens to be an engineered network.

1.1.2. Environment

Environment is a two-tuple that is syntactically a subset of an agent:

$$Environment = \langle State_e, Process_e \rangle$$

The important feature of this definition of the environment (and one often overlooked in laboratory environments) is that the environment itself is active. It has its own Process that can change its State, independent of the actions of its embedded agents.

$Input_i$ and $Output_i$ of the various agents are coupled to elements of $State_e$, but the environment does not distinguish which elements of $State_e$ are so coupled. That distinction depends on the agents that exist at any moment and the capabilities of their sensors and effectors. The lack of a distinguished Input and Output means that the environment, unlike an agent, is unbounded. The fact that environment, like agent, has State and Process means that putting a boundary and associated Input and Output values around an environment (and its associated agents) yields a higher-level agent. Thus this model lends itself naturally to aggregating low-level agents into higher-level ones.

1.1.3. Coupling

The exact nature of the coupling depends on how we model State and Process. Table 1 summarizes two approaches that have been taken.

Table 1: Two Models of State and Process

Model	Discrete-Event Dynamical Systems	Time-Based Dynamical Systems
State	Discrete (symbols)	Continuous (real numbers)
Processing	Symbol manipulation, e.g., -- Rewrite rules -- Transition tables	Partial differential equations or difference equations
Progress	Discrete; event-based	Discrete (integer time) or Continuous (real time)
Time & State	Independent (clock speed)	Coupled (physical laws)
Communities	Computer Science, AI	Physics, Mathematical Ecology

Computer scientists, particularly those with a background in symbolic computation (as in artificial intelligence), naturally model systems as though their dynamics were governed by discrete events. Canonical models of such systems include rewrite grammars and automata driven by transition tables (such as a finite-state automaton or a Turing machine). Such a system leaps instantaneously from one discrete state to another on the basis of events that are not necessarily anchored to a numerical time parameter. As a result, there is no intrinsic coupling between numerically valued time and the state of the model. The system yields the same result when run on a fast computer as on a slow one, and only an outside observer with access to a real-time clock can tell the difference.

Students of the natural sciences naturally model systems as time-based dynamical systems, in which state variables can be real numbers and thus vary continuously, and in which the progress of the system is based not on the sequence of a series of events, but on a numerical time parameter. In such a system, processing speed is intimately linked to the physics of the domain, and any attempt to change the “clock speed” of the process can be expected to change the qualitative behavior of the system.

It can be argued that the most appropriate model to use for a system depends on the level of analysis. A digital computer, for example, is built on time-based continuous quantum wave functions at the subatomic level. When one moves to the level of holes and free electrons in a semiconductor, a discrete-event model is more appropriate, and continues to dominate through the level of gates, logic circuits, and program execution. However, it becomes inefficient to consider the state space of a very large network of computers as discrete, and time-based models with continuous state sometimes emerge for analyzing such systems.

Most work on artificial agents identifies itself with the artificial intelligence community, which views the problem of building an agent as one of artificial cognition. AI “orthodoxy” advocates the Physical Symbol System hypothesis, classically expounded by [18], which asserts that a cognitive system is a discrete-event dynamical system.[1] As a result of this orthodoxy, most research on agents takes a discrete-event approach to modeling agents (and the environment, if the researcher models it explicitly at all). By contrast, most researchers who focus their attention on the environments (other than computational) in which agents must perform use time-based models.

An understanding of the coupling between agent and environment must consider four cases, corresponding to the interacting distinctions {Discrete-Event, Time-Based} x {Agent, Environment}. The simplest cases are when the dynamics of both agent and environment are modeled the same way (a situation we describe as homodynamic). Additional complications arise when the system is heterodynamic, with discrete-event dynamics for one element and time-based or mixed dynamics for another. Agent systems based on electronic environments (e.g., knowbots and personal assistants) tend to be homodynamic (with Discrete Event dynamics for both agents and environment), while industrial applications tend to be heterodynamic (with Discrete Event agents and Time-Based environment).

If all communication is mediated through an environment that agents share, it follows that any agent action that changes that environment is potentially communicative (that is, visible to other agents and having an effect on their activity).

1.2. All Agent Activity is Non-Deterministic

For the balance of this discussion, we will assume Discrete Event agents. Such agents are in general individually deterministic, in the sense that they consist of sequential programs of explicitly defined computational operations. An apparent exception is offered by agents that are aggregates of lower-level agents, but such combinations must use some form of communication to integrate their parts, so that the resulting non-determinacy can be traced again to communication. A deterministic agent can control its own state changes. However, it cannot control the effect of its actions in the environment or on other agents. There are three reasons for this. The first two, autonomy and concurrency, can occur in either homodynamic or heterodynamic systems. The third, impotence, is a specific problem in heterodynamic systems.

Autonomy.—In a system is composed of autonomous agents and an active environment, no one agent has unilateral control over the rest of the system. Furthermore, with limited rationality, no one agent can be sure that it has perfect and complete knowledge of the rest of the system. As a result, it can never assume that other agents will do what it tells them, or that the environment will change as expected when it activates its effectors. In a closed system in which all agents and their environment are designed together, designers may include mechanisms to enforce control relationships among agents or guarantee the adequacy of each agent's models of its surroundings, but such a strategy forfeits the decoupling among system components that is one of the great promises of agent technology.

---

[1] Recently, [16] have articulated the view of a small but growing group of researchers who assert that cognition is a time-based dynamical system. For purposes of contrast with the Physical Symbol System hypothesis, they call their approach the Dynamical System Hypothesis. This moniker, while memorable, is a little misleading, since both approaches use dynamical (that is, state-based) systems.

Concurrency.—It has been recognized for some time that when different processes concurrently manipulate the same state variables, the result is nondeterministic [12]. [4] show how this basic insight is often overlooked in discussions of agent planning and interaction. The resulting (unrealistic) identity between an agent’s actions and the resulting change in the environment contributes to classical AI conundrums such as the Frame Problem. As with autonomy, in a closed system one might enforce locking mechanisms such as semaphores to ensure that concurrently executing agents do not interfere with one another. Again, such an approach forfeits the vision of open systems, in which agents from different sources interoperate without advanced special arrangements. In this case, agents cannot assume that their actions have the expected effect.

Impotence.—Heterodynamic systems frequently arise in industrial applications, with Discrete Event agents and a Time-Based environment. This case complicates the coupling between agent and environment. In addition to a mapping between the state spaces of agent and environment, the agent’s sensors must provide an energy sink and an analog-to-digital conversion to respond to energy flows from the environment, and its actuators must provide an energy source and a digital-to-analog conversion in order to provide energy flows to the environment. Now the coupling takes the form of an energy flow. Because state values in the environment can vary continuously, the change in a variable resulting from such a flow may be infinitesimal, and an agent’s input may not be able to detect it. Conversely, an agent’s output may not be powerful enough to overcome competing forces impinging on the same variables it is addressing. Thus the agent may not see what it thinks it is looking at, and may not have an effect where it is exerting action.

In addition to these three reasons that an agent cannot be sure of the outcome of its actions, limited rationality implies that it can’t sense its way out of the dilemma by maintaining a comprehensive model of the world.

The result of autonomy, concurrency, impotence, and limited rationality is that an agent cannot assume that its actions have the desired effect. This observation leads naturally to the third Provocative Claim:

1.3. All Agent Activity is Interactive

Because an agent cannot know the results of its actions, it must interact with its environment, and thus (at least indirectly) with any other agents with whom it shares the environment. At least two paradigms are common for managing this interaction.

Simple agents use some variety of state machine to coordinate the agent’s actions with its inputs. A display example is Ferber’s BRIC formalism, which uses Petri nets to describe agent behaviors [3].

More complex agents may use various planning techniques to reason their way from their Beliefs about the state of the world to the state they Desire, thus producing Intentions to guide their immediate actions [6].

2. What is and isn’t Negotiation?

The workshop introduction defines negotiation as “interactive communication for arranging activity.” In the light of the Provocative Claims (PC’s), this definition applies to virtually any action by an agent embedded in a shared environment, and thus offers no useful distinction. I'll sketch the argument, then propose how to fix the definition so that in some cases it makes a useful distinction. However, for sufficiently complex agents (including those with capabilities anticipated by many agent researchers today), even this improved definition cannot overcome the deconstruction implicit in the PC's.

2.1. Inadequacy of the Simple Definition

To see the inadequacy of the simple definition, consider four successively complex mechanisms for agent interaction, summarized in Table 2. The "common sense" meaning of negotiation is usually restricted to the two most complex mechanisms in the list: fixed protocols (which are typically implemented using state machines) and open-ended conversations (which require planning capability on the part of the agents). However, even voting, command, and reactive systems fall under the workshop definition when the PC's are taken into account. Thus "negotiation" ceases to be a useful category for distinguishing different kinds of agents.

Table 2: Levels of Agent Interaction

Protocol	Description	Minimal Language
Reactive	Sense, then Act	Environment
Command	Master agent sends unilateral instructions to servants	Symbolic
Voting	One-shot quantitative statement of interest	Currency
Fixed Protocol	Back-and-forth; symbolic & quantitative	Messages & Protocol (Contract Net)
Conversation	Arbitrary interchange	Speech Acts (statement, request, commitment; command; e.g., KQML)

Voting.—A paradigm example of a voting system is the LMS system for short-interval scheduling of a semiconductor fab [5]. The problem is to determine which batch of wafers will next gain access to a bottleneck tool. Separate agents vote on each batch. Each agent represents a different perspective, and votes on a range of 0 (don't want that batch) to 1 (must have that batch).

- How far is the batch behind schedule?
- What is the current inventory status of the chips that the batch will produce?
- What downstream tools are currently starved for input, and what types of wafers do they need to keep busy?
- What is the current setup of this tool, and which batch will best utilize it?

The agents vote, an umpire combines their votes and makes a decision. At first glance this is a unidirectional communication. However, in formulating its vote, each agent draws on state information that results from the collective outcome of previous rounds of voting. For example, if the agent watching inventory status loses a number of rounds, the strength of its vote will increase in later rounds as inventory drops. If it wins several, its later votes will drop as inventory grows. The dominant PC here is the first. Because the agents share a common environment (including, in the case of the example, the inventory), their actions inevitably respond to one another.

Command.—Some early hierarchically-structured agent systems such as YAMS [14] and BOSS [7] follow the NASREM architecture [1] in which higher-level agents unilaterally command their subordinates. At first glance, such a pattern seems diametrically opposed to the notion of “negotiation.” However, because of nondeterminism (PC 2), subordinates in this architecture must report back on their status to their superiors. Thus even command-based communications is interactive. The status report is not just a vestige; if it is unsatisfactory or if it does not arrive within a designated time period, the master agent typically takes corrective action, such as repeating the command. If the status report indicates the subordinate is unable to execute the command (or if a time-out suggests that the subordinate is off-line), the master may very reasonably select a different subordinate. In this case the interaction becomes indistinguishable from a rudimentary contract net, which is undeniably an implementation of negotiation.

Reaction.—How about reactive systems, in which agents simply sense their local environment and then act, without any conventional messaging behavior at all? The fact of a shared environment means that these acts are communicative in effect (PC 1), and reactive agents do in fact interact based on these shared effects (PC 3). The parade example is insect ecology, much of which is occupied with deconstructing apparently complex interactions into simple environmental reactions. An outside observer might very well attribute the coordinated behavior of a set of reactive agents to explicit negotiation, and be surprised to learn the simple underlying mechanisms involved. An example closer to home to the DAI/MAS community is the pursuit problem, originally introduced to the community by Miro Benda at the 1986 DAI Workshop at Twin Lights Manor [2]. For several years this problem attracted a variety of research approaches drawing from a wide range of sophisticated AI techniques, including negotiation in the usual sense of the word. A review of previous work is available in [8], which reports a reactive mechanism that outperforms previous more sophisticated approaches. While the degree of Korf’s success may owe something to the simplicity of the particular instantiation of the pursuit problem under consideration [10], the general moral remains, that extremely simple modes of interaction with the environment can yield results comparable with what looks at first like more sophisticated inter-agent transactions.

2.2. Tightening the Definition

All the objections that I have been able to conceive against the argument above boil down to a question of intentionality. Intentionality is not a part of the dictionary definitions of “communication” cited above. They, like the Provocative Claims, focus on information transmission and its effects, not the intentions of the participants. One might reformulate the definition something like this: “Negotiation is intentionally interactive communication among agents.” (It’s not clear to me that “arranging activity” contributes anything to the original definition. Agents that don’t act aren’t very interesting, and virtually anything that an agent does contributes to arranging its activity.) Such a reformulation presents several ambiguities that we must anticipate.

First, to what is intentionality being applied: interaction or communication? The restriction really works only if it is applied to both. The agent’s activity must be intended to be communicative to other agents, and there must be the intention that those other agents will interact and communicate in return.

Second, whose intentionality is being invoked: the agent’s, or the agent designer’s? Designer intentionality is not enough to overcome the deconstruction. The designer must recognize the kinds of back-door interactions implied by the PC’s above, or run the risk of exactly the kind of anomalous system behavior that we experienced with YAMS. YAMS negotiated in the “full” sense, using a contract net, but the designer of a set of reactive agents would still need to exercise intention in managing their (implicit) interactions through the environment. So such agents would enjoy designer intentionality, but still not meet our intuitive threshold for negotiation. We must focus on the intentionality of the agents themselves.

Third, what does it mean for a computer program to have intentions? Behavioristic definitions (“An agent intends whenever an observer would attribute intention to it”) don’t help, since they can’t make the distinction we need in this case. So we are left with definitions of “intention” that point us to some explicit knowledge structure and associated inferencing that manages the interactive communication. These structures differ from the higher-level structures that a successful agent must have for dealing with all interactions with the environment, and with the implicit interactions that they support, because they explicitly take account of the other agents involved in the interaction (intentionality of communication as well as of interaction). Most commonly, these structures will form an explicit model of the other partners in the negotiation that is used to managing its course. Alternatively but perhaps less usefully, they may distinguished the channels by which they access the environment (e.g., whatever I do over the internet is negotiation, but whatever I do through my D/A converter is not).

We tend to think of sophistication in negotiation as a hallmark of relatively sophisticated agents. One ironic consequence of this definition is that the more sophisticated an agent’s models of its environment become, the less useful the concept of “negotiation” is as distinguishing among alternative forms of interaction. One can clearly identify the agent’s intentional model in a simple physical agent that has a state machine to talk to other agents and a D/A interface to actuators and sensors. However, in a super-smart agent that analyzes the changes it sees in the environment, imputes some of them to other agents, and acts accordingly, there may be no objective way to draw the line between negotiative and non-negotiative activity. Even our improved definition no longer makes a useful distinction, and the deconstruction persists.

3. What are some Opportunities for Negotiation Research?

The first opportunity is improving the definition so that we have a crisper notion of the subject of discussion. Even with the explicit addition of intentionality, I’m still not happy that we have a definition that makes a useful distinction between what is and what is not negotiation.

Once we recognize that communication takes place implicitly through the environment, we can begin to make more effective explicit use of the environment as a medium for deliberate, interactive communication. In many industrial applications, information can be communicated through physical changes to the environment much more reliably and unambiguously than through digital channels. One might capture this insight as a corollary to the principle that “the world is its own best model” by arguing that in many cases “the world is its own best network” as well. If our objective is to do research into negotiation, then the simplicity of implicit interaction through the environment is a threat to our funding. But if our objective is to achieve coordinated agent interaction for some application, we may find these alternative approaches very attractive from an engineering perspective.

Even in digital environments, we may be able to learn something from the nature of environment-based interactions. In physical domains, the latter rely on the interactions of quantitative fields rather than transmission of symbolic structures. As I have argued in [15], field-based interaction is essential for overcoming the pressures exerted by second-law forces toward increased entropy in communities of agents.

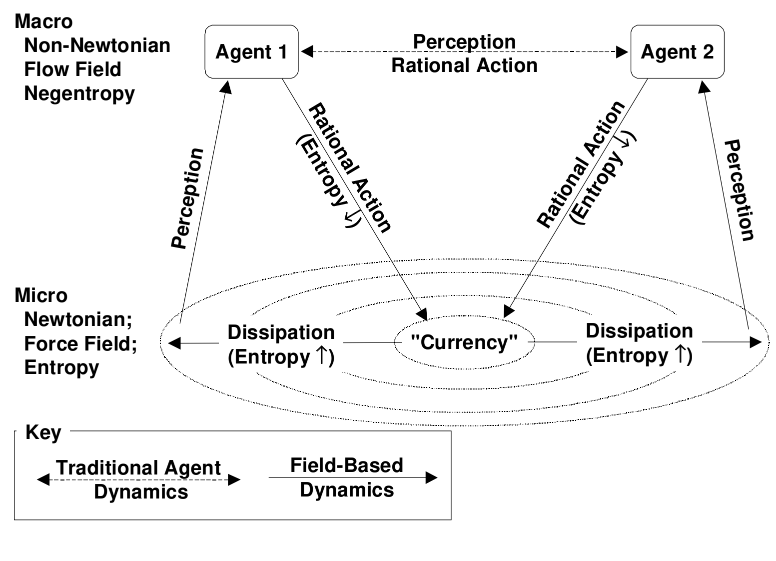

Figure 1: Macro Organization through Micro Dissipation

The Second Law of Thermodynamics warns that large agent communities tend to disorder, not organization. Natural systems can organize themselves at the macro level because their actions are coupled to a dissipative process at a micro level. The resulting "entropy leak" drains disorder away from the level where useful work is done to one where it won't interfere with the system's function. The dissipation actually supports coordination among agents, as they *perceive* and orient themselves to the flow field generated by dissipation and reinforce that field by their *rational action*, as shown by the solid lines in Figure 1 [9].

For example, insect colonies leak entropy into the Brownian dissipation of pheromone molecules, generating a pheromone field that guides individuals. In markets, entrepreneurs perceive the flow of money and orient themselves to it, resulting in the self-organization of structures such as supply chains and geographic economic centers.

Traditional mechanisms of agent coordination ignore the micro level completely, as agents perceive and act directly on one another (dashed line in Figure 1). This approach provides no escape for entropy that naturally builds up in the system. Figure 1 suggests that the answer is not imposing a separate control mechanism alongside existing coordination models, but linking agents through the micro environment so that perception and action serve both to coordinate multiple agents and to control disorder in the overall system.

[19] offer an extreme example of such a mechanism, in which agents generate typed attractive and repulsive fields that vary over time, with the effect of scheduling the execution of manufacturing tasks on machines. Priority-based interaction mechanisms as in [17] are another useful example. In fact, I'd argue that the current interest in market-based mechanisms, with its emphasis on quantitative communications, is a healthy instance of this kind of mechanism, and one that we should actively seek to extend.

4. What are some Pitfalls in Negotiation Research?

The major pitfall that emerges from this paper is the danger of unanticipated "side-bar" interactions through the environment. The effectiveness and proven performance characteristics of a negotiation mechanism demonstrated in a laboratory setting can be completely vitiated by these back-door interactions. Agent designers must learn to think of agent interaction at a more encompassing level than just conventional communications if they are to avoid the kind of breakdown we experienced in YAMS.

A counterpart of the opportunity of using the environment for interaction is the pitfall of ignoring it and as a result constructing a system that is more complex and fragile than it needs to be. Again, sound engineering requires taking a system view of interaction, not one restricted to an academically convenient subset. Our objective is agent coordination, not negotiation *per se*, and we should be open to any mechanisms that can help with that objective.

While I have not focused on it in this discussion, the dynamics of negotiation (and of agent interaction in general) is a critical question that is only now beginning to be addressed. We need assurances that proposed mechanisms will not lead to undesirable oscillations or chaotic interchanges, but that a course of interaction (whether intentional or implicit) will converge to the overall results desired by the system designer.

References

[1] J. S. Albus, H. G. McCain, and R. Lumia. NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM). NBS Technical Note 1235, National Bureau of Standards, Gaithersburg, MD, 1987. *A classic example of a hierarchical control architecture, in which each node (or “agent”) generates commands for agents at lower levels and synthesizes status for those above it. This model dominates traditional control systems, but is extremely rigid, and (as our experience with YAMS showed) can lead to communications bottlenecks as information works its way up and back down the hierarchy in order to move between siblings or cousins in the hierarchy.*

[2] M. Benda, V. Jagannathan, and R. Dodhiawalla. On optimal cooperation of knowledge sources. BCS-G2010-28, Boeing Computer Services, Bellevue, WA, 1985. *The working paper presented at the Twin Lights DAI Workshop that proposed the Pursuit Problem as a challenge problem for agent research. At that time, the problem was conceived as a good demonstration environment for fairly complex planning and coordination techniques. This expectation was overturned by Korf’s work, referenced later in this bibliography.*

[3] J. Ferber. *Les systèmes multi-agents: vers une intelligence collective*. Paris, France, InterEditions, 1995. *A significant textbook on multi-agent systems, covering both coarse-grained (“BDI”) and fine-grained agent architectures (Eco-problem solving). It outlines the BRIC model for defining the behavior of agents using Petri nets. Happily, it is scheduled to appear this fall from Addison-Wesley in English.*

[4] J. Ferber and J.-P. Müller. Influences and Reactions: a Model of Situated Multiagent Systems. *Proceedings of Second International Conference on Multi-Agent Systems (ICMAS-96)*, pages 72-79, 1996. *A fundamental analysis of the problem of Concurrency as a driver of system-level non-determinism even when individual agents are deterministic. The authors propose a formalism for modeling the interaction of several agents in a shared environment, and discuss how their perspective can help address the Frame Problem. The basic (and essential idea) is that an agent can’t assume that the world changed just because it tried to change it. Tthe world may have its own countervailing processes, or other agents may be undoing what it is trying to do.*

[5] K. Fordyce and G. G. Sullivan. Logistics Management System (LMS): Integrating Decision Technologies for Dispatch Scheduling in Semiconductor Manufacturing. In M. B.

Morgan, Editor, *Intelligent Scheduling*, pages 473-516. Morgan Kaufman Publishers, Inc., San Francisco, 1994. *A system that uses reactive agents to do short-interval scheduling in a semiconductor fab. In this discussion, LMS serves as a minimalist architecture that nevertheless can be viewed as using negotiation. Although the successive stations in the fab do not communicate directly with one another, their decisions clearly do depend on one another, as communicated through their impact on a shared environment.*

[6] A. Haddadi and K. Sundermeyer. Belief-Desire-Intention Agent Architectures. In G. M. P. O'Hare and N. R. Jennings, Editors, *Foundations of Distributed Artificial Intelligence*, pages 169-185. John Wiley, New York, NY, 1996. *A useful summary statement of the BDI architecture; used in this discussion as a maximalist architecture.*

[7] J. E. Hynynen. BOSS: An Artifically Intelligent System for Distributed Factory Scheduling. In F. Kimura and A. Rolstadas, Editors, *Computer Applications in Production and Engineering*, pages 667-77. Elsevier, 1989. *An example of an agent-based system that organizes agent interaction around a hierarchical structure, following the classical architecture outlined by Albus. Within each node in the hierarchy is a blackboard architecture with functional agents.*

[8] R. E. Korf. A Simple Solution to Pursuit Games. *Proceedings of Eleventh International Workshop on Distributed Artificial Intelligence, 1992. A solution to the pursuit problem proposed by Benda et al. that does not use methods usually thought of as "negotiation." It is interesting because previous solutions to the problem often did draw on such techniques. Thus this article shows the possibility of unexpectedly simple solutions, and urges us not to assume more complexity than necessary.*

[9] P. N. Kugler and M. T. Turvey. *Information, Natural Law, and the Self-Assembly of Rhythmic Movement.* Lawrence Erlbaum, 1987. *This volume introduced me to the basic principles behind Figure 1 above, which embodies several key concepts: 1) We need to worry explicitly about second-law pressures to disorder in multi-agent systems. 2) A good way to handle this problem is to "leak" the undesired entropy to a micro level to balance the overall entropy books. 3) This leakage is not just a way to get rid of unwanted disorder, but in fact can provide a field to which agents orient themselves, thus positively supporting self-organization.*

[10] M. Manela and J. A. Campbell. Designing Good Pursuit Problems as Testbeds for Distributed AI: A Novel Application of Genetic Algorithms. *Proceedings of 5th European Workshop on Modeling Autonomous Agents in a Multi-Agent World, MAAMAW '93*, pages 231-252, Springer, 1995. *A qualification to Korf's results, showing that the relative performances of reactive and more sophisticated techniques may depend on the detailed structure of the particular pursuit problem. This paper, Benda et al., and Korf taken together form an excellent example of the kind of research dialog that we need to construct and maintain in order to develop a good understanding of coordination mechanisms (including negotiation) and where they should be used.*

[11] Merriam-Webster. WWWebster Dictionary. www.m-w.com/netdict.htm, 1998. *The source for the "dictionary" definition of "negotiation" used in this paper.*

[12] R. Milner. Elements of Interaction. *Communications of the ACM*, 36(1 (January)):78-89, 1993. *An analysis of how concurrent deterministic processes can yield a non-deterministic*

system. While this paper does not directly discuss negotiation techniques, it does clearly describe and document some of the challenges that negotiation confronts.

[13] J. P. Müller. *The Design of Intelligent Agents*. Berlin, Springer, 1996. *Jörg’s analysis of agents takes into account explicitly the fact that they interact through a shared environment. He thus anticipates some of the issues outlined in this paper.*

[14] H. V. D. Parunak. Manufacturing Experience with the Contract Net. In M. N. Huhns, Editor, *Distributed Artificial Intelligence*, pages 285-310. Pitman, London, 1987. *Describes the YAMS system, an early application of negotiation to the problem of shop floor control. This paper documents an instance in which a system whose communication protocol had been proven deadlock-proof nevertheless deadlocked in operation. The agents exchanged not only messages but also physical parts, and it turned out that the movement of a part from one machine to another provided information leakage that in effect extended the “official” communication protocol between the agents, thus invalidating the liveness proof. This anecdote has become an important part of our awareness that discussion of communications must not be limited to digital channels.*

[15] H. V. D. Parunak. ’Go to the Ant’: Engineering Principles from Natural Agent Systems. *Annals of Operations Research*, 75:69-101, 1997. Available at http://www.erim.org/~van/gotoant.ps. *An analysis of naturally-occurring fine-grained agent systems for principles that can be applied to synthetic agent systems. Includes a discussion of environmentally-mediated agent interaction and homodynamic vs. heterodynamic systems. This document is my attempt to outline in one place many of the insights that lie behind the “synthetic ecosystem” approach to multi-agent systems.*

[16] R. F. Port and T. vanGelder, eds. *Mind as Motion: Explorations in the Dynamics of Cognition*. Cambridge, MA, MIT Press, 1995. *A bold challenge to the dominant model in classical AI, the Physical Symbol System Hypothesis. Port and vanGelder and their contributing authors propose that cognition is fundamentally a time-based dynamical system rather than an event-based set of rewrite rules. Their examples include not only muscular motor control, but also such traditionally symbolic domains as symbol recognition and grammatical analysis. The discussion is overdrawn, perhaps for the sake of making the point, but extremely reading for anybody working in traditional AI.*

[17] R. Sikora and M. J. Shaw. Coordination Mechanisms for Multi-Agent Manufacturing Systems: Applications to Integrated Manufacturing Scheduling. *IEEE Transactions on Engineering Management*, 44(2 (May)):175-187, 1997. *An application of a priority-based coordination mechanism that is one example of how field-based techniques can be used to achieve results otherwise attempted through more complex negotiation protocols.*

[18] H. A. Simon. *The Sciences of the Artificial*. 2nd Edition. Cambridge, MA, MIT Press, 1981. *Classical statement of the “physical symbol system hypothesis,” which asserts that cognition is best modeled as a system of discrete rewrite rules over an alphabet of symbols. [I read this years ago, and don’t have personal copy; it strikes me on writing this note that this may not be where Simon articulates the PSSH. Can any readers either confirm, or point me to a better source? My purpose here is simply to document the hypothesis in order to contrast it with that of Port and vanGelder.*

[19] J. Vaario and K. Ueda. An emergent modelling method for dynamic scheduling. *Journal of Intelligent Manufacturing*, 9(2 (April)):129-140, 1998. *This paper is an extreme example of applying field-based coordination to a real-world problem (in this case, real-time scheduling of a manufacturing facility). Their approach is to surround transport mechanisms and machines with synthetic fields of shapes and intensities designed to attract transporters to their destinations while avoiding collisions. While the scheme is clearly not yet mature enough for physical application, it does capture well the spirit of environmentally mediated interaction.*

