

A Design Taxonomy of Multi-Agent Interactions

H. Van Dyke Parunak¹, Sven Brueckner¹, Mitch Fleischer¹, James Odell²

¹Altarum, 3520 Green Ct Ste 300
Ann Arbor, MI 48105 USA, +1-734-302-4600
{van.parunak, sven.brueckner, mitch.fleischer}@altarum.org

² James Odell Associates, 3646 Huron River Drive
Ann Arbor, MI 48103, +1-734-994-0833
email@jamesodell.com

Abstract. Agent interactions are frequently characterized as “coherent,” “collaborative,” “cooperative,” “competitive,” or “coordinated.” These terms specialize the more foundational category of “correlation,” which can be measured by the joint information of a system. “Congruence” is orthogonal to the others, reflecting the degree to which correlation and its specializations satisfy user requirements. A taxonomy of these mechanisms can guide the design of multi-agent interaction. Lack of correlation is sometimes necessary, and requires the use of formal stochasticity.

1 Introduction

Designers of multi-agent systems need to discuss agents’ joint action in a disciplined way that current usage does not support. Some circles emphasize cooperation as the dominant theme (e.g., the title of [23] or the subtitle of [46]), but often the system designer does not care whether the agents cooperate or contend, so long as their behavior is coordinated in a certain way. Market mechanisms achieve coordination through mechanisms that are at least competitive and sometimes contentious. Both contention and cooperation presume a cognitive model that some architectures do not satisfy. Other terms for agent interaction include “coherence” and “collaboration” Taking advantage of the pervasive use of the Latin *co-* and *con-* in the English lexicon, we refer to them collectively (including nominal, verbal, and adjectival forms) as “Co-X,” and from this point capitalize them.

Paper titles from [6, 18], ICMAS’95-00, Agents’97, 98, 00, and 01, and AAMAS’02 illustrate the proliferation of terms describing agent interactions. From 1981 through 1993,

“Cooperate”¹ accounted for eight of the twelve “Co-X” terms in titles, “Coordinate” for three, and “Coherent” for one. “Collaborate” appeared in 1994, and “Compete” in 1995. For 1994 through 2002, 120 Co-X terms were used, with “Coordinate” at 41%, followed by “Cooperate” at 38%, “Collaborate” at 16%, and “Compete” and “Cohere” at 2.5% each. The vocabulary is growing beyond “Cooperation.” Yet, review of this literature shows that there is little agreement on defining the various members of Co-X. Formal analyses of teams and social behavior (e.g., [11, 12, 13, 43]) helpfully refine concepts such as common knowledge, joint intentions, commitments, obligations, and social norms. These and related concepts help achieve the behavior that Co-X describes, but the usage of the Co-X terms themselves remains intuitive and sometimes inconsistent.

A taxonomy of these and other terms can enable a more precise description of how agents interact. This precision is a critical contribution to agent-oriented software engineering, since it provides a foundation for improved system specifications and improved communication among users, designers, and implementers of multi-agent systems. Under our definitions, the terms are neither a mutually exclusive spanning set such that every agent-based system belongs to exactly one term in the set (“categories”) nor an orthogonal set each of whose terms can be applied to all agent-based systems (“perspectives”). A formal taxonomy requires a complete structure of both categories and perspectives [31], but at this point we claim only, in the words of one reader of an earlier draft, “a nice start.”

Section 2 defines the most general term, “Correlation,” statistically, without assumptions about either the internal structure of the agents or the relative centralization or decentralization of their behavior. Section 3 defines “Coordination” as Correlation that results from information flows among agents. When these flows result from individual agent intentions, we speak of “Cooperation” and “Competition,” discussed in Section 4, along with “Collaboration,” which is the intersection of Cooperation and “Conversation” (a specialization of Coordination). Section 5 discusses the “Congruence” of group behavior with system-level intentions. Section 6 returns to the fundamental notion of Correlation and examines ways in which it can and cannot be avoided. Section 7 offers a summary.

¹ Including the noun “Cooperation” and the adjective “Cooperative.” When we refer to one grammatical form of a given word, we intend our observations to apply to the others as well.

2 Correlation: Behavioral Joint Information

The most generic description of what agents do together is their joint information, otherwise known as their correlation entropy, joint entropy, or mutual entropy [2]. This quantity can be determined empirically, without access either to the internal structure of the agents or to the broader system within which they are embedded. A set of agents with positive joint information is “Correlated.” In some cases, such empirical observations may be sufficient to impute cognition to agents whose internal structure is unknown [8, 21, 34].

Joint information can be computed for many aspects of the agents. We are interested in agent behaviors, so we compute it over agent actions. At each time step, each agent (indexed by i) can execute one of n_i actions $\{a_{i1}, a_{i2}, \dots, a_{in_i}\}$. Let p_{ij} be the probability that agent i executes action a_{ij} . We estimate this quantity by maintaining a time series of the agent’s last k actions, counting the actions of each type, and dividing by k . One measure of the agent’s behavior over time is its Shannon entropy, $H(a_i) = - \sum_{j=1}^{n_i} p_{ij} \log_2 p_{ij}$. This

standard definition makes no assumptions about the independence of the elements in the set that generates the p_{ij} . It is simply an empirical characterization of their relative prevalence. (If k is small compared with agent lifetime, one can index p_{ij} and thus H with time t , referencing the window of length k centered at t .) We can characterize the entropy of the overall system in terms of the various combinations of actions of individual agents. E.g., for two agents, the maximum total number of system actions is $n_1 * n_2$, p is indexed over joint actions, and the system entropy is $H(a_1, a_2) = - \sum_{j=1}^{n_1 * n_2} p_j \log_2 p_j$ (again, indexable by t).

System entropy is subadditive, $H(a_1, a_2) \leq H(a_1) + H(a_2)$. Equality obtains when individual agent behaviors are statistically independent, and fails when they are dependent. The difference $I(a_1:a_2) \equiv H(a_1) + H(a_2) - H(a_1, a_2)$ is the correlation, mutual, or joint entropy, or the joint information. The latter term avoids connotations of disorder implicit in “entropy.” Agent behaviors in a system with high joint information are more Correlated with one another, and in that sense more orderly, than when joint information is low.

Agents are Correlated when their actions depend statistically on those of other agents. It does not matter at this level whether this Correlation results from information flows among the agents or between them and a central controller, or whether its roots are cognitive or subcognitive. Correlation manifests itself in an increase of the system’s joint information, and we propose using this quantity as a measure of system Correlation. (Thus, one system has a higher Correlation than another just when the joint information of the

first is greater than that of the second.) This perspective permits us to distinguish the *fact* of Correlation from the *mechanisms* used to achieve it.

For example, suppose that each of two agents needs access to a widget to perform its duties; that there are two widgets available, and that each agent accesses each widget half of the time. Then the probability that agent a_1 is accessing widget 1 is $p_{1,1} = 0.5$, as are $p_{1,2}$, $p_{2,1}$, and $p_{2,2}$. Assume that a widget works better when only one agent is using it at a time. Each individual agent entropy $H(a_i)$ is $-2 * 0.5 \log_2(0.5) = 1.0$. At the system level, there are four possible joint actions: both agents accessing widget 1, both accessing widget 2, a_1 accessing widget 1 and a_2 widget 2, and *vice versa*. If the agents do not Coordinate their activities, then each of these four possibilities is equally likely, with probability 0.25, and the system entropy $H(a_1, a_2)$ is $-4 * 0.25 \log_2(0.25) = 2 = \sum H(a_i)$, so the joint information is 0. Now assume the agents' behaviors are Correlated (by whatever means) so that they avoid the joint actions in which they both choose the same widget. Now there are only two system actions, each with probability 0.5, and the system entropy is 1, less than the sum of the agent entropies. The difference, 1, is the joint information between the agents.

We illustrate joint information in a model of multi-agent resource allocation, the minority game [36]. At each time step each of the N agents in the population (where N is odd) seeks to allocate itself to one of two resources. Each agent receives a point each time it reaches the less-occupied (minority) resource. The system goal is to maximize the total points awarded to the entire population (equivalently, to minimize the variance in the population of either resource). The agents have access to a time series identifying which resource was in the minority at each past cycle. In each turn of the game, the agents use the last m entries in this time series to choose the resource they will access on the next time step. The quantity $z = 2^m/N$ reflects the normalized size of the strategy space available to the agents.

Figure 1 plots the normalized variance σ^2/N as a function of z . Low variance reflects high system-level payoff, so desirable behavior is at the minimum of this curve, where $z \approx 0.34$. The dashed line shows the performance if all agents made random choices. The minimum is a spin-glass phase transition, discussed in more detail in [26, 38]. As z decreases below this point, the system performance degenerates until the agents are doing

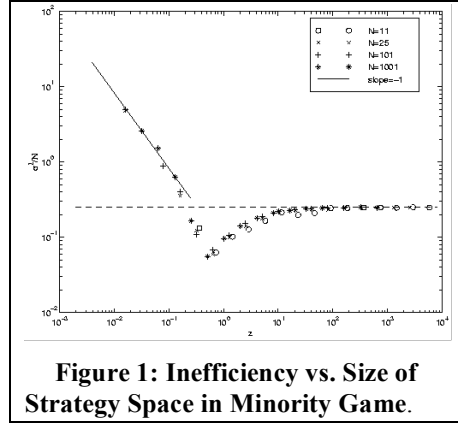


Figure 1: Inefficiency vs. Size of Strategy Space in Minority Game.

worse than with random choices. In this region, the time series of minority groups [25] show “herding” behavior. Relatively few distinct strategies are available for small m , with high probability that agents’ decisions will overlap. Above the phase transition, the agents do better than random, but as z increases, performance approaches the random limit as an asymptote.

Consider a set of 61 agents. Each agent’s entropy is computed from its probability of choosing resource 0 or 1 at each step. The entropy of the system is computed from the probability of a specific vector of 61 individual agent choices at a time step. Thus the size of the system’s state space is $2^{61} \approx 10^{18}$. Reasonable experimental runs with this system are on the order of 10^4 to 10^6 steps, far too short to estimate probabilities over this state space. Instead, we focus on subsystems of six agents each. Such a subsystem has a state space of 2^6 , and we can reliably estimate probabilities with experiments of $10^4 \approx 2^{13}$ steps. Each run of 5001 steps lets us look at ten subsystems of six agents each, and we conduct thirty runs in all. Figure 2 shows the Correlation, measured by the joint information. This figure has three important features, corresponding to the three regions of Figure 1.

1. The Correlation is highest for low m , consistent with the analysis in [25] showing herding behavior in this region. Within the low region, the mean value appears to increase from $m = 1$ to $m = 2$ before declining for $m > 2$, but given the size of the standard deviation in region, the most that can be said is that the Correlation is comparable for these two values. There are few distinguishable strategies available to the agents for low m , resulting in higher Correlation among their behaviors.
2. High m is associated with low Correlation, corresponding to the region where agent decisions approach the random limit. Thus the general shape of the curve is logistic: low slope for low and high m , and steep slope in between.
3. There is a deviation from this general shape between $m = 4$ and $m = 5$, the region that corresponds to the phase transition (which would be at $m = 4.37$, though our experiments do not sample this point).

This illustration of joint information has several lessons for designers of multi-agent systems.

- Joint information can be computed for a realistic system, offering a basis for metrics that will enable designers to specify degrees of Correlation. Since Correlation is the foundation on which the other members of Co-X are constructed, this formalism is an important tool for comparing different systems in terms of the degree of joint activity that their members achieve.

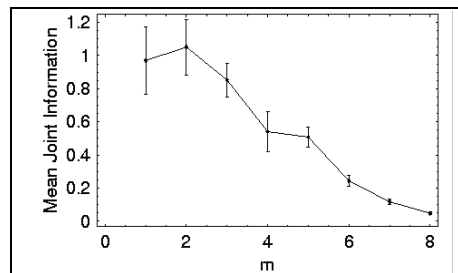


Figure 2: Joint Information in the Minority Game.—Error bars mark one standard deviation.

- When estimating joint information from observations of a system (as opposed to computing it analytically from a system specification), attention must be paid to sampling issues.
- Joint information is not the same as performance. Figure 1, which estimates overall system performance, has a very different shape from Figure 2, which shows Correlation. System designers must recognize this distinction, discussed in more detail in Section 5.

3 Coordination: Communication

Perhaps the most commonly used member of Co-X is “Coordination.” It is prominent in the ACM computing classification system under I.2.11, “Distributed Artificial Intelligence” [1] (paired with “Coherence,” discussed under “Congruence” below). “Correlation” describes simply the fact of statistical non-independence among agent behaviors, while “Coordination” implies a causal process. Correlation can emerge among randomly generated numbers as a statistical fluke, but when it arises from a causal process, that process involves communication, that is, information flow between an individual agent and its environment. The environment, in turn, is everything outside the individual agent’s boundary. The options for this flow stem from the contents of this environment, which include both other agents and environmental state variables. (A side effect of this definition is to require redefinition of either “Coordination” or “communication” in paper titles of the form, “Coordination without communication” [3, 15, 16, 40].)

The environment includes other agents, not only software agents but also human stakeholders, system designers, and conventional computer systems. The relationship between two such agents will be one of two types, depending on the state of the agents and the rules of the system (expressed, for example, in the agents’ roles [35] and the protocols in which they participate). When the agents can say “No” to one another within the rules of the system, they are “peer agents.” When one of them (say agent A) can say “No” to the other (B), but B cannot say “No” to A, we call A the “distinguished agent” and B the “subordinate.” The relationship between two agents may be fairly fixed (for example, the relationship between a human programmer and her software agent). Or it may vary over time (as when peer agents negotiate a work plan that calls for one of them to supervise the other, resulting in a distinguished-subordinate relationship during execution). These concepts can be developed more formally through dependency and autonomy theory [10, 30].

The environment has state variables that the agents can sense, and may support its own processes that couple its state variables and cause them to change over time. Environmental variables are of two types. The values of *endogenous* variables change over time

Table 1: Categories of Communication

		Topology of Inter-Agent Relationships	
		Centralized (between Distinguished and Subordinate agents)	Decentralized (among Peer agents)
Information Flow	Direct (messages between agents)	<i>Construction</i> (Build-Time) <i>Command</i> (Run-time)	<i>Conversation</i>
	Indirect (non-message interaction)	<i>Constraint</i>	<i>Stigmergy</i> ² (generic) <i>Competition</i> (limited resources)

depending on the actions of peer agents. The values of *exogenous* variables, such as sunspot frequency, change independent of the actions of peer agents, but may result from actions of the distinguished agent, and are often scripted by the system designer.

In such an environment, Correlation mechanisms can be either centralized or decentralized. In addition, the information flows involved may be either direct from agent to agent (ignoring those aspects of the environment that make up the communication system) or indirect (mediated through explicitly modeled environmental state variables). Table 1 reflects these categories, which offer a design template for agent interaction.

Centralized mechanisms for Correlation all involve communication between the distinguished agent and its subordinates. This flow may be direct (when the distinguished agent Constructs or Commands the subordinates) or indirect (when the distinguished agent Constrains the subordinates by manipulating exogenous environmental variables visible to the subordinates). In Correlation through Command, used commonly in robot soccer, holonic manufacturing, and some simulation applications, agents behave much like objects, executing methods invoked by incoming messages. The focal point algorithm advocated by [15] and the common utility functions implicit in [16] both rely on Construction (common programming). In indirect centralized mechanisms, subordinates jointly sense changes in a shared exogenous environmental variable. The variable’s dynamics are independent of agent actions, so it cannot move information between subordinates. But it may serve as a synchronizing signal that Correlates the agents’ actions. The experimenter who configures targets and obstacles in an experimental testbed is Constraining the subordinates, supporting Correlation through indirect centralized action.

Decentralized mechanisms for Correlation all involve communication among peers. Most negotiation research focuses on direct peer-to-peer information flows (“Conversation”). Indirect decentralized flows occur when peers make and sense changes to endoge-

² Unlike other terms in Co-X, “Stigmergy” does not begin with “Co,” but the term is too well established in the research community to replace it with an alliterative alternative.

nous environmental variables. This class of Coordination is called “stigmergy,” [17], from the Greek words *stigma* “sign” and *ergon* “work”: the work performed by agents in the environment guides their later actions. Such techniques are common in biological distributed decentralized systems such as insect colonies [32]. A common form of stigmergy is resource Competition, which occurs when agents seek access to limited resources. For example, if one agent consumes part of a shared resource, other agents accessing that resource will observe its reduced availability, and may modify their behavior accordingly. Even less directly, if one agent increases its use of resource A, thereby increasing its maintenance requirements, the loading on maintenance resource B may increase, decreasing its availability to other agents who would like to access B directly. In the latter case, environmental processes contribute to the dynamics of the state variables involved. (We reserve “Competition” for resource Competition as a subset of Stigmergy. For the more generic opposite of “Cooperation,” we prefer “Contention,” discussed below.)

We became aware of the active role that the environment plays in negotiation when experimenting with an instance of the contract net for manufacturing control [29]. After carefully proving that our protocol was deadlock-free, we ran it on a physical control system, and it promptly deadlocked. The negotiation in question concerned the movement of a physical part from one workstation to another. Our analysis of the protocol neglected the movement of the physical part itself. This movement conveyed information between the two workstations, and thus between the software agents that represented them. It represented an undocumented extension of our protocol, one that invalidated our proof and caused the system to deadlock. The arrival of the part at the receiving workstation gave that workstation information about the state of the system that it would not otherwise have had, namely, that the part had been delivered.

Traditionally, the study of negotiation focuses on Coordination by means of information flow directly from one agent to another. The mantra of situated robotics that “the world is its own best model” [7] suggests that the problem domain may deserve a more prominent role in the process. There are several motives for understanding the role of the environment in Coordination, and learning to exploit it where possible.

- It supports open, heterogeneous societies of agents. The environment is by definition accessible to the agents that are negotiating about it. Any agent that wishes to deal with the domain must be able to sense and manipulate it. Thus the physics of the environment define common standards for agent interaction, in contrast with the more arbitrary standards programmers can impose on direct agent-to-agent communication.
- It integrates and reflects the state and dynamics of the overall problem-solving process at a global level that is only imperfectly visible in any individual agent’s internal model. In particular, it captures high-order interaction effects that may escape the notice of any individual agent or *a priori* model maintained by an individual agent. For instance, assume agents A, B, C, and D are all interested in resource \square , but A and B

know only of each other, as do C and D. The load on resource integrates information about the demands of all the agents that would otherwise not be available to them.

- It embeds domain constraints (e.g., resource limitations) directly in the reasoning process, without the need to identify and model them in advance.

A stock market illustrates the importance of information flows mediated by endogenous environmental variables. It affects both stock traders and business executives, in different ways. Traders (at least those who obey SEC regulations) do not communicate directly to determine which shares each will buy and sell. But when a trader offers for sale a share in one company, the offer tends to depress that company's share price, making the company more attractive to potential buyers. Thus information flows between traders through the stock market without Conversation. In contrast, business executives rely extensively on Conversation in reaching contracts with their customers and suppliers. However, they must also pay attention to indirect information flows, including those through the same stock market. For example, if a supplier's stock price drops precipitously, the supplier may not be able to raise needed capital, and in spite of its explicit promises in a negotiation, it may not be able to fulfill its obligations.

The Minority Game is an excellent example of Stigmergy, and of Competition in particular, and we discuss its implication for indirect Coordination further in [36].

These mechanisms reflect Coordination mechanisms recognized by sociologists in organizational design. One prominent discussion [27] distinguishes five such mechanisms:

1. Mutual adjustment, informal communication among workers, corresponds to the Direct Decentralized quadrant of Table 1, which we call "Conversation."
2. Direct supervision is our "Command," which represents the real-time portion of our "Direct Centralized" quadrant.
3. Standardization of work processes (e.g., setting up a work station on an assembly line) is adjusting the environment to Constrain agents to behave in a certain way, and thus corresponds to our "Indirect Centralized" quadrant.
4. Standardization of outputs insures that intermediate outputs from one worker can be input to the next, enabling "stigmergy," our "Indirect Decentralized" quadrant.
5. Standardization of skills and knowledge trains workers to behave in Coordinated ways. This is our "Construction," in the "Direct Centralized" quadrant of the table.

4 Cooperation³ and Contention: Intent

Correlation is an empirical concept that requires no knowledge about agents' internal structure or outward organization. The focus on communication emphasized by *Coordination* requires attention to inter-agent issues, but leaves agents' internal logic undefined. Cooperation and Contention involve the agents' intentions. For example, behaviors of traders in a commodity market are highly Correlated, resulting from information flows among them (thus Coordination). Are they Cooperating or Contending? Two traders bidding for the same commodity might be Contending (each seeking to wrest control from the other), Cooperating (pumping the price up to increase the value of their current holdings), or simply Competing (in the sense defined in Section 3). The difference can only be resolved by determining their intent (compare [24]).

A necessary condition for Cooperation is the existence across the Cooperating agents of joint intentions (e.g., [11]). Similarly, contention suggests one agent's intention to frustrate another. To our knowledge, this notion of "antagonistic intent" has not been formalized, but could be along the same lines as joint intention. We do not require intention for Competition, respecting the common use of the term for agents seeking common limited resources without harboring malice toward one another.

The definition of Cooperation and Contention as Correlation driven by agent intent has two important implications.

First, imputing Cooperation and Contention to agents also requires imputing cognition to them. This requirement is easy for Cognitive architectures (e.g., SOAR [28] or BDI [37]), which imitate human cognition. It is less direct for Behavioristic agents. Such agents, inspired by work in artificial life [33], are "black boxes," defined only by their outward behavior, and their internal programming makes no claims to imitate cognition. It is possible to impute cognition to such agents in a disciplined way [34], but doing so would require observing other behaviors beyond the specific actions to be classified as Cooperation or Contention, in order to deduce their intentions toward one another.

Second, neither Cooperation nor Contention requires direct decentralized communication ("Conversation," Table 1), but might result from centralized design-time information flows or interactions through the environment. The special case when agents both Converse and Cooperate is "Collaboration"; "Coalition" describes the resulting state of affairs.

³ Axtell [4] notes that game theoreticians would reverse our definitions of "coordination" and "cooperation." Our definitions are more in line with the usage in the MAS community. The exact words used are much less important than precision in distinguishing the processes involved.

5 Congruence and Coherence: Usefulness

None of the modes of interaction discussed thus far is necessarily desirable. Consider the simple problem of Correlated access by two agents to two widgets considered in Section 1. If the agents avoid choosing the same widget, they are Correlated, achieving a joint information of 1. They will be just as Correlated if they always choose the same widget, but in this case productivity would be lower in the Correlated system than in the random one. Similar examples can be constructed to illustrate that increased Coordination, Cooperation, and Competition do not always result in more productive systems.

The crucial insight here, and one of great importance for agent-oriented software engineers, is that systems can be associated with goals at two levels: the system, and the individual agents. (Cognitive agents reason explicitly about these goals, while in behavioristic agents they are imposed by the agent designer, but they are still agent-level goals.) Categories such as Contention and Cooperation take into account individual agent goals, but not system goals. We propose “Congruence” to characterize the degree to which the pattern of agent interactions (at any level from Correlation through Contention and Cooperation) satisfies (“is Congruent with”) system-level goals. “System-level” is critical. For example, in an e-commerce system, each individual agent may have a different user with different goals (e.g., increased market share vs. short-term profit). Congruence deals, not with the conformity of individual agents to the goals of their respective users, but with the conformity of the system as a whole to its system-level goals (e.g., bounded transaction times, information availability, and transaction security). The relation among the agents that yields Congruence is “Coherence” (Figure 3), a term that without definition in the ACM Computing Classification [1]. (Durfee et al. [14] define “Coherent” as “well-coordinated.”)

System-level goals can arise in two different ways. In an engineered system, they are defined by the system’s creators [41]. We term these “top-down goals.” In a cognitive multi-agent system, they can also emerge from agent interactions [42], whether through democratic processes or by the imposition on other agents of the individual goals of an agent that has gained a controlling position in the society. These goals are thus “emergent goals.” Several points need to be made.

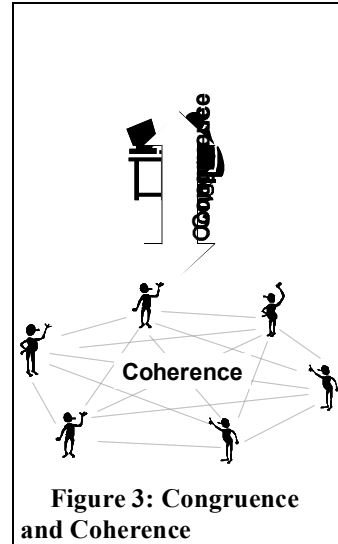


Figure 3: Congruence and Coherence

1. Congruence does not presuppose either peer-to-peer information flows or individual agent intent. It may exist with any form of Correlation.
2. Congruence cannot be defined for systems that do not have system goals. Like intentions, emergent goals are most naturally associated with cognitive agents. Behavioristic agents can be congruent in three ways. First, their creator may define their system-level goal. Second, in a larger system that includes both non-cognitive and cognitive agents, the cognitive agents may define emergent system goals for the entire system, and do their best to impose them on the non-cognitive portion of the system. Third, such goals may be imputed to them per [34].
3. A system may have conflicting goals, arising in populations of non-cognitive agents from inconsistency in the designer's goals, in populations of cognitive agents from tensions among different emergent processes, and in created systems of cognitive agents from a disjunction between the designer's goals and goals that emerge within the population. Congruence is defined only with respect to a specified goal or set of goals.

System-level goals are needed to *define* Congruence, but whether or not they *affect* it depends on whether individual agents can sense and respond to them. Emergent goals that do change the behavior of the system exemplify “downward causation” [39].

The minority game exemplifies a top-down system goal (maximum total points awarded across the population). This goal is not downwardly causative, because the agents do not know of it or reason about it, and the system is Congruent only in the vicinity of the phase transition. Changes in Correlation (Figure 2) reflect the coarse structure of system performance (Figure 1), but are not statistically correlated with them. In particular, the highest level of Correlation (and thus Coordination) occurs for low m , while the system is most Congruent and the agents most Coherent for intermediate levels of m .

System-level goals (e.g., “norms,” “conventions,” and “obligations”) are the subject of considerable study ([13] and references). Whatever such theory one adopts will set a standard against which to assess Congruence.

6 Anticorrelation

The various members of Co-X are all refinements of Correlation. Our discussion of Congruence and Coherence suggests that more Correlation is not always good thing.

1. The agent system might be in Contention with an adversary that could exploit observed regularities in its performance. In such cases, the system should avoid regularities, and appear as though it were made up of statistically independent entities.
2. Due perhaps to similarities in their internal coding, the agents may tend to “run into” each other in the problem space, and need to spread out to do their job effectively.

3. The agent system may use some form of weak search (e.g., particle swarm optimization [22] or evolutionary computation [19]) to search the space of system behaviors. Such mechanisms assume ergodicity: they depend on the system's dynamics to sample the state space, and Correlation leads to under-represented regions of the state space.

Agents can anticorrelate if each makes its decisions via random processes (either at a central controller or in each agent). Can they be more deliberate, basing decisions on either peer-to-peer or master-slave information flows? We can make two observations.

1. Any system with at least two Correlated agents is Correlated. Let $A = \{a_i\}$ be the set of peers, $B = \{b_i\} \subset A$ the subset that is Correlated, $H(A)$ the entropy of the entire system, and $H(a_i)$ the entropy of the i th peer. Perfect anticorrelation requires $H(A) = \sum H(a_i)$ (summing over the elements of A). A component of this sum is $H(B)$. But B is Correlated, so $H(B) < \sum H(b_i)$. Each b_i contributes less than $H(b_i)$ to $H(B)$ and thus to $H(A)$, so $H(A) < \sum H(a_i)$, and the entire system is Correlated.
2. Any set of more than one anticorrelating agents must use a random process in their decision-making to achieve anticorrelation. Assume otherwise. Then their actions are a deterministic function either of a non-random central signal or of observations (direct or indirect) of one another's behavior. But then each agent's behavior is not statistically independent of the actions of the other agents, and the system will be Correlated.

Observation 1 makes it unlikely that MAS engineers will ever deal with perfectly anticorrelated systems. Correlation wants to happen. If agents are behaving in any way other than randomly, their aggregate behavior will reflect it. In other words, emergent behavior is ubiquitous. This behavior may not be Congruent (e.g., herding in financial markets), but it will be Correlated. Rather than viewing emergent behavior as a threat to be suppressed by constraining the behavior of individual agents so that the system exhibits only a subset of its total potential behavior [9, 20, 45], software engineers should understand the mechanisms that drive emergence so that we can harness it for productive use.

Observation 2 emphasizes the importance of stochasticity as an element of multi-agent systems. If we want agents to spread out through their joint state space, we can do no better than to have them flip coins. In the parlance of statistical mechanics, such a device provides the "symmetry breaking" that avoids undesirable Correlation. A system's level of organization is inversely proportional to its level of symmetry [5], and random variations among agents is a powerful way to introduce differences that can be amplified by agent interactions to yield self-organization. Many techniques of swarm intelligence [32] include a stochastic element, including random walks in ant path planning, Fermi functions in swarm sorting algorithms, and stochastic face-offs in the emergence of organization in *Polistes* wasps. Elsewhere [34] we exhibit a simple artificial agent whose performance is dramatically improved by addition of random noise.

7 Conclusion

Agents do things together. Clear discussions of what they do, and effective designs of how to do it, require precision in the terms we use to describe joint behavior. Such improved precision is particularly critical for software engineers responsible for specifying, designing, constructing, and deploying multi-agent systems.

The fundamental characteristic is *Correlation*, defined as nonzero joint information over a population of agents. Agent Correlation is a purely behavioral notion. It requires knowledge only of the observed actions of the agents. If we admit other sorts of knowledge, we can refine it in three orthogonal ways.

Coordination is Correlation with a focus on the information flow that enables it, and six different flavors can be distinguished: Conversation, Construction, Command, Constraint, Stigmergy, and Competition. The main distinctions are whether the information flow is centralized or peer-to-peer, and direct or indirect. Thus Coordination implies a particular architecture between agents, but is silent about their internal processing.

Cooperation and *contention* modulate Correlation by the intent of individual agents. Cooperation requires joint intentions, while Contention requires an intention on the part of one agent to frustrate another. Both concepts impute cognition to the participating agents (thus requiring special care in the case of behavioristic agents), but they are silent regarding the inter-agent architecture, and thus independent of Coordination. A system with both Conversation (direct peer-peer communications) and Cooperation (joint intent on the part of the individual agents) exhibits *Collaboration*, which results in *Coalitions* of agents.

Congruence measures alignment of an agent system with a system-level goal, which may be defined either endogenously or exogenously. It is independent of both inter-agent and intra-agent architecture. *Coherence* is the relation among agents that yields Congruence. Congruence is not necessarily a monotonic function of Correlation. Sometimes increased Correlation (or Coordination, or Cooperation) may yield lower Congruence.

More disciplined attention to these distinctions will enable more effective specification, design, and deployment of multi-agent systems. This analysis suggests a number of directions for future work.

- This preliminary taxonomy should be extended. For example, it is fruitful to consider temporal distinctions in the ways agents work together, such as synchrony vs. asynchrony [44].
- The taxonomy provides a basis on which to review current agent engineering models and modeling languages (e.g., AUML) for completeness and expressivity.⁴

⁴ We are grateful to an anonymous referee for suggesting this and the following point.

- The next step after defining the members of Co-X is to discuss how to design agents that achieve them. Such a discussion could extend the preliminary suggestions in this paper (e.g., our taxonomy of communication types) with specific patterns and criteria for selecting among them.

Acknowledgments

This work is partly supported by DARPA ANTS and NA3TIVE under contracts F30602-99-C-0202 and N00014-02-C-0458 to Altarum. The views and conclusions in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the US Government.

References

- [1] ACM. The ACM Computing Classification System [1998 Version]. 1998. HTML, <http://www.acm.org/class/1998/>.
- [2] C. Adami. *Introduction to Artificial Life*. New York, NY, Springer Telos, 1998.
- [3] R. C. Arkin. Cooperation without Communication: Multiagent Schema-Based Robot Navigation. *Journal of Robotic Systems*, 9(3):351-364, 1992.
- [4] R. Axtell. Personal communication 2002.
- [5] P. Ball. *The Self-Made Tapestry: Pattern Formation in Nature*. Princeton, NJ, Princeton University Press, 1996.
- [6] A. H. Bond and L. Gasser, Editors. *Readings in Distributed Artificial Intelligence*. San Mateo, CA, Morgan Kaufmann, 1988.
- [7] R. A. Brooks. Intelligence Without Representation. *Artificial Intelligence*, 47:139-59, 1991.
- [8] B. Browning, G. A. Kaminka, and M. M. Veloso. Principled Monitoring of Distributed Agents for Detection of Coordination Failures. In *Proceedings of Distributed Autonomous Robotic Systems (DARS-02)*, 2002.
- [9] S. Bussmann. Agent-Oriented Programming of Manufacturing Control Tasks. In *Proceedings of Third International Conference on Multi-Agent Systems (ICMAS'98)*, 57-63, IEEE Computer Society, 1998.
- [10] C. Castelfranchi. Founding Agent's 'Autonomy' on Dependence Theory. In *Proceedings of 14th European Conference on Artificial Intelligence*, 353-357, IOS Press, 2000.
- [11] P. Cohen and H. J. Levesque. Teamwork. Technical Report Technote 504, SRI International, Menlo Park, CA, 1991.

- [12] F. Dignum and B. v. Linder. Modelling social agents: Communication as actions. In M. Wooldridge, J. Muller, and N. Jennings, Editors, *Intelligent Agents III*, vol. 1193, *LNAI*, 205-218. Springer-Verlag, New York, NY, 1997.
- [13] F. Dignum, D. Morley, E. A. Sonenberg, and L. Cavedon. Towards Socially Sophisticated BDI Agents. In *Proceedings of Fourth International Conference on MultiAgent Systems (ICMAS'2000)*, 111-118, IEEE Computer Society, 2000.
- [14] E. H. Durfee, V. R. Lesser, and D. D. Corkill. Coherent Cooperation among Communicating Problem Solvers. *IEEE Transactions on Computers*, C-36:1275-1291, 1987.
- [15] M. Fenster, S. Kraus, and J. S. Rosenschein. Coordination without Communication: Experimental Validation of Focal Point Techniques. In *Proceedings of International Conference on Multi-Agent Systems (ICMAS'95)*, 102-108, AAAI, 1995.
- [16] M. R. Genesereth, M. Ginsburg, and J. S. Rosenschein. Cooperation without Communication. In *Proceedings of National Conference on Artificial Intelligence (AAAI'86)*, 51-57, AAAI, 1986.
- [17] P.-P. Grassé. La Reconstruction du nid et les Coordinations Inter-Individuelles chez *Belligositermes Natalensis et Cubitermes sp.* La théorie de la Stigmergie: Essai d'interprétation du Comportement des Termites Constructeurs. *Insectes Sociaux*, 6:41-84, 1959.
- [18] M. N. Huhns and M. P. Singh, Editors. *Readings in Agents*. San Francisco, CA, Morgan Kaufmann, 1998.
- [19] C. Jacob. *Illustrating Evolutionary Computation With Mathematica*. San Francisco, Morgan Kaufmann, 2001.
- [20] N. R. Jennings. On Agent-Based Software Engineering. *Artificial Intelligence*, 117:277-296, 2000.
- [21] G. A. Kaminka, M. Fidanboyly, A. Chang, and M. Veloso. Learning the Sequential Behavior of Teams from Observations. In *Proceedings of RoboCup Symposium*, 2002.
- [22] J. Kennedy, R. C. Eberhart, and Y. Shi. *Swarm Intelligence*. San Francisco, Morgan Kaufmann, 2001.
- [23] V. Lesser and D. D. Corkill. Functionally accurate, cooperative distributed systems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-11:81-96, 1981.
- [24] A. Lux and D. Steiner. Understanding Cooperation: An Agent's Perspective. In *Proceedings of First International Conference on Multi-Agent Systems (ICMAS'95)*, 261-268, MIT and AAAI, 1995.
- [25] R. Manuca, Y. Li, R. Riolo, and R. Savit. The Structure of Adaptive Competition in Minority Games. Program for the Study of Complex Systems, University of Michigan, Ann Arbor, MI, 1998. URL <http://www.pscs.umich.edu/RESEARCH/pscs-tr.html>.
- [26] M. Marsili, D. Challet, and R. Zecchina. Exact solution of a modied El Farol's bar problem: Efficiency and the role of market impact. 1999. PDF File, http://tt.lanl.gov/PS_cache/cond-mat/pdf/9908/9908480.pdf.
- [27] H. Mintzberg. *Structure in Fives: Designing Effective Organizations*. Englewood Cliffs, NJ, Prentice-Hall, 1993.
- [28] A. Newell. *Unified Theories of Cognition*. Cambridge, MA, Harvard University Press, 1990.
- [29] H. V. D. Parunak. Manufacturing Experience with the Contract Net. In M. N. Huhns, Editor, *Distributed Artificial Intelligence*, 285-310. Pitman, London, 1987.
- [30] H. V. D. Parunak. Distributed AI and Manufacturing Control: Some Issues and Insights. In Y. Demazeau and J.-P. Müller, Editors, *Decentralized AI*, 81-104. North-Holland, 1990.
- [31] H. V. D. Parunak. Hypercubes Grow on Trees (and Other Observations from the Land of Hypersets). In *Proceedings of The Fifth ACM Conference on Hypertext*, 73-81, ACM, 1993.

- [32] H. V. D. Parunak. 'Go to the Ant': Engineering Principles from Natural Agent Systems. *Annals of Operations Research*, 75:69-101, 1997.
- [33] H. V. D. Parunak, S. Brueckner, and J. Sauter. ERIM's Approach to Fine-Grained Agents. In *Proceedings of NASA/JPL Workshop on Radical Agent Concepts (WRAC'2001)*, Forthcoming, 2001.
- [34] H. V. D. Parunak and S. A. Brueckner. Imputing Agent Cognition from Dynamics. In *Proceedings of Autonomous Agents and Multi-Agent Systems (AAMAS 2003)*, (submitted), 2003.
- [35] H. V. D. Parunak and J. Odell. Representing Social Structures in UML for Agent-Oriented Software Engineering. In *Proceedings of Workshop on Agent-Oriented Software Engineering*, 17-24, 2001.
- [36] H. V. D. Parunak, R. Savit, S. A. Brueckner, and J. Sauter. Experiments in Indirect Negotiation. In *Proceedings of The AAAI Fall 2001 Symposium on Negotiation Methods for Autonomous Cooperative Systems*, 2001.
- [37] A. S. Rao and M. P. Georgeff. Modeling Rational Agents within a BDI Architecture. In *Proceedings of International Conference on Principles of Knowledge Representation and Reasoning (KR-91)*, 473-484, Morgan Kaufman, 1991.
- [38] R. Savit, R. Manuca, and R. Riolo. Adaptive Competition, Market Efficiency, Phase Transitions and Spin-Glasses. PSCS-97-12-001, University of Michigan, Program for the Study of Complex Systems, Ann Arbor, MI, 1997. URL <http://xxx.lanl.gov/abs/adap-org/9712006>.
- [39] R. K. Sawyer. Simulating Emergence and Downward Causation in Small Groups. In *Proceedings of Multi-Agent-Based Simulation (MABS'2000)*, 49-67, Springer, 2000.
- [40] S. Sen, M. Sekaran, and J. Hale. Learning to Coordinate Without Sharing Information. In *Proceedings of National Conference on Artificial Intelligence (AAAI'94)*, 426-431, AAAI, 1994.
- [41] Y. Shoham and M. Tennenholtz. On Social Laws for Artificial Agent Societies: Off-Line Design. *Artificial Intelligence*, 73:231-252, 1995.
- [42] Y. Shoham and M. Tennenholtz. On the Emergence of Social Conventions: modeling, analysis and simulations. *Journal of Artificial Intelligence*, 94(1-2):139-166, 1997.
- [43] G. Tidhar, E. A. Sonenberg, and A. S. Rao. On Team Knowledge and Common Knowledge. In *Proceedings of Third International Conference on Multi-Agent Systems*, 301-308, IEEE Computer Society, 1998.
- [44] D. Weyns and T. Holvoet. Synchronous versus Asynchronous Collaboration in Situated Multi-Agent Systems. In *Proceedings of Autonomous Agents and Multi-Agent Systems (AAMAS 2003)*, (submitted), 2003.
- [45] M. J. Wooldridge and N. R. Jennings. Pitfalls of Agent-Oriented Development. In *Proceedings of 2nd Int. Conf. on Autonomous Agents (Agents-98)*, 385-391, 1998.
- [46] M. Yokoo. *Distributed Constraint Satisfaction: Foundations of Cooperation in Multi-Agent Systems*. Berlin, Springer, 2001.