Multiagent Systems as Spheres of Commitment

Munindar P. Singh*
Department of Computer Science
North Carolina State University
Raleigh, NC 27695-8206, USA

singh@ncsu.edu

Abstract

The notion of commitment is central to understanding agents and multiagent systems. A related notion arises in databases and distributed computing. We compare the two notions, and propose a synthesis, which combines the best properties of both. Our synthesis involves a generalization of *spheres of control*. We show how multiagent systems can function as generalized spheres of commitment, addressing the issue of structure, and providing context for commitments. This yields a generic approach to commitment that can handle discharge and cancelation naturally.

*This work is supported by the NCSU College of Engineering, the National Science Foundation under grants IRI-9529179 and IRI-9624425, and IBM corporation.

1 Introduction

The notion of *social commitment* is central to DAI. This is the familiar notion where an agent makes a commitment to another agent (with certain nuances that we discuss below). This paper does not discuss *psychological commitment* [Singh, 1996], so "commitment" means "social commitment" throughout. Commitments have drawn much research attention because they are an important abstraction for characterizing, understanding, analyzing, and designing multiagent systems. Commitments help coordinate and structure multiagent systems to achieve coherence in their actions.

Commitments also arise in distributed databases. However, traditional distributed databases implement a highly restrictive form of commitment—procedurally hard-wired and irrevocable. Modern applications of databases, which involve heterogeneous databases and programs, and human collaboration, do not fit the traditional mold. Some of these applications have been addressed using agent-based techniques, e.g., [Wittig, 1992; Singh & Huhns, 1994]; others with advanced database techniques, e.g., [Bukhres & Elmagarmid, 1996]; and still others by combining in organizational techniques, e.g., [Papazoglou *et al.*, 1992]. Such applications require greater organizational flexibility reflected in more flexible forms of commitment.

Multiagent systems are finding increasing application in heterogeneous and open information environments. As multiagent systems are deployed in serious applications, there will be increased expectations of robustness, involving harder guarantees of the atomicity of actions, their durability, and recoverability in case of system failure. Present-generation multiagent systems do of meet the standards of traditional databases in this regard. Our ongoing research program is to develop abstractions through which robust and flexible multiagent systems can be developed.

As part of that greater goal, we propose a more sophisticated notion of commitment that marries ideas from DAI with those from *spheres of control*, which is a conceptual approach (introduced in databases) for structuring activities. This paper

- discusses commitment and spheres of control in databases
- discusses commitment in multiagent systems, and its connections with group structure
- relates the above concepts to yield a formalization of commitment that explicitly considers context and cancelation.

Section 2 introduces the database notions of commitment. Section 3 discusses multiagent commitment, and relates it to the structure of multiagent systems. Section 4 formalizes commitment to capture several useful properties. Section 5 concludes with a review of the pertinent literature.

2 Database Notions of Activity and Commitment

Some of the most significant applications of multiagent systems are in heterogeneous, open information environments. For this reason and to better understand the intuitions developed in traditional distributed database settings, it is instructive to review commitment as it is implemented in those settings.

2.1 Traditional Database Transactions

Traditional transactions are computations that satisfy a number of useful properties, in particular the so-called ACID properties [Gray & Reuter, 1993].

- *atomicity:* all or none of a transaction happens
- *consistency:* a transaction preserves the consistency of the database
- *isolation:* intermediate results of a transaction are not visible externally
- *durability:* when a transaction concludes successfully, its effects are permanent.

If the individual transactions are programmed correctly, the system guarantees consistency for any arbitrary concurrent mix of transactions. Atomicity is essential to ensure that the integrity of distributed data is preserved. Consequently, the actions or subtransactions that constitute a transaction must either (a) all happen, thereby transforming the database from a consistent state to a new consistent state, or (b) each fail to happen, thereby leaving the database in its original (consistent) state.

Example 1 Consider a banking transaction that executes at sites A and B, and transfers money from an account in site A to an account in site B. Assume one subtransaction per site. The two subtransactions must agree to terminate successfully or fail (failure may be caused at some site because of an underlying problem, or user intervention). Otherwise, money would be spuriously created or lost! ▌

Atomic commitment protocols ensure that the sites achieve this agreement. The most famous of these protocols is two-phase commit (2PC). In 2PC, there is a voting phase in which each process votes Yes or No. The votes are collected by a coordinator, who initiates the decision phase. If all processes voted Yes, the coordinator announces a Yes decision, otherwise the coordinator announces a No decision.

Two observations follow immediately from the above. One, the above phenomenon really is commitment, since the processes guarantee that they will behave as the coordinator decided. However, this commitment is a one-shot affair. The processes agree on whatever they are doing, do it, and move on (typically all terminate). The commitment they exhibit is irrevocable, but since the interaction is short-lived, that cannot cause significant problems. Two, the commitment is represented trivially. There is typically a flag set inside each process. The commitment aspects are provided externally, through an agreement among the various programmers. This agreement ensures that processes will respect the decision of the coordinator, and once they commit to a decision, they will not change their mind. Thus each site knows that the other sites will behave as it expects.

The traditional approach is indeed the best when dealing with hard-coded systems in which the ACID properties are essential. However, in many modern applications, the ACID properties may be either

- unnecessary—e.g., it might unnecessary to maintain consistency of some street traffic management data, which will be overwritten with new data anyway;
- impossible to satisfy—e.g., it is impossible to model a travel reservation activity involving multiple airlines and hotels as a transaction, because the different companies won't let you run atomic commit protocols; or
- undesirable—e.g., it is obviously undesirable to model activities of two collaborating humans as transactions, because then they would (by definition) be isolated, i.e., uncollaborative!

For the above reasons, superior primitives are needed for modeling composite activities, and the commitments among their components.

2.2 Spheres of Control

Spheres of control (SoC), which sought to characterize activities more generally than database transactions, were proposed about two decades ago [Davies, 1978]. The database community is seeing a resurgence of interest in SoCs as the limitations of traditional transactions are obstructing their application in open environments [Gray & Reuter, 1993, pages 174–180].

Intuitively, SoCs attempt to contain the effects of an action as long as there might be a necessity to undo them. A result can be released only when it is established that it is correct (and will remain correct). If a result may have to be undone, a SoC must be set up outside of the activities that consume the result, so the result may be undone. SoC are defined in terms of three interacting concerns:

- *Process atomicity:* This determines what actions are included in the given unit of atomicity. All or none of these actions are performed.
- *Process control:* This determines which data items are owned by a process—these cannot be modified by others.

- *Process commitment:* This identifies a specific function that determines the modified value of each data item.

We now apply SoCs on Example 1 of section 2.1.

Example 2 The traditional (transactional) approach of Example 1 corresponds to a single SoC that includes both accounts: (a) changes to the accounts occur atomically; (b) the SoC controls both accounts, so nothing else can modify them while it is executing; and (c) the SoC commits both accounts, so their new values cannot be used by any other activity until they have been committed. ▮

Example 3 Following Example 1, define separate SoCs for each account. Thus partial, inconsistent results become visible—e.g., when one account has been debited, but the other has not yet been credited. This is a temporary glitch. More importantly, we also lose process atomicity—if exactly one SoC succeeds, the results will be permanently inconsistent. The SoC solution is to create an SoC enclosing the other two SoCs. The outer SoC determines if an inner SoC needs to be undone, and undoes it, thereby ensuring process atomicity. The outer SoC must retain enough of the context in order to undo actions properly. Undoing a committed change can cause further activities to be invalidated, requiring them to be undone, possibly causing further effects. This phenomenon is sometimes called *cascading aborts* and, while undesirable from a performance standpoint, it might be required for correctness. ▮

Conceptually, SoCs are related to a number of extended transaction models that have been proposed [Elmagarmid, 1992; Bukhres & Elmagarmid, 1996]. SoCs are thus a useful idea, and significantly more powerful than traditional database transactions. However, they are not formalized to any significant extent, and have proved difficult to implement [Gray & Reuter, 1993, page 180]—they presuppose elaborate mechanisms for undoing the effects of committed activities. We submit that SoCs can find a natural home in appropriately structured multiagent systems. SoCs would be more practical in the kinds of deliberative coarse-grained reasoning situations common in multiagent system applications.

3 Commitments in Multiagent Systems

SoCs, while including the germ of a good idea, end up being too rigid in some respects and vague in others. What we like about them is the idea of building enclosing spheres within which the control, commitment, and atomicity requirements of the enclosed spheres can be relaxed. We adopt this idea in our treatment of commitments in multiagent systems. We do not repeat here the numerous reasons for studying commitments in multiagent systems, but show how they can be married with SoCs to give yield a powerful means of structuring computations.

3.1 Commitments and Structure

Commitments have an intimate relationship with the structure of multiagent systems. We previously identified two relevant principles [Singh, 1991]:

- *Heterogeneity:* Systems are composed of a diverse mix of agents and other systems; the members may have different knowledge, know-how, intentions, etc. In some cases, they may not be aware of each others' existence, or know each others' names.

- *Monolithicity:* Despite their heterogeneity, multiagent systems can be viewed as if they were single agents, who perform actions and have beliefs, intentions, commitments, and so on. Groups such as teams, armies, nations, and corporations may all be (and are in fact) profitably treated as agents in their own right.

The structure of a group is given by the constraints on the interactions among its members. The interactions can be *social* or *ecological*, as defined next.

- *Social agency:* Communications are the primary form of social interaction. Constraints specify how various communications are effected, and the range of effects they might have. These rely upon commitments among the communicating agents. For example, promises ordinarily bring into effect a commitment by the speaker to the hearer; directives presuppose a commitment by the hearer to do as told; assertives commit the speaker to the statement expressed; permissions make the speaker committed to allowing the relevant condition to hold; prohibitions presuppose a commitment by the hearer to preventing the relevant condition from holding. Notice that promises and assertives bring about the relevant commitments, even if the speaker is *insincere*! The speaker is subject to censure for having lied [Castelfranchi, 1993]. More generally, interaction protocols, e.g., for various kinds of negotiation, may be defined so that the agents have the relevant commitments at appropriate times.

- *Ecological agency:* The interactions may also arise because of infrastructural reasons. For example, agents might act in a coordinated manner because of subtle properties of the environment they inhabit. Indeed, an agent might obtain some information from another agent because of their resource conflicts. However, these interactions differ from the communicative ones in lacking the corresponding commitments.

This paper concentrates on social agency. In our framework, agents can be individuals or groups, who may be recursively composed of groups. A useful paradigm is of an agent being committed to a group that contains him. This enables capturing several common social situations, e.g., when several agents simultaneously enter into related commitments with each other.

We use the term *commiter* to refer to the agent that makes a commitment, and the term *commitee* (not "committee") to refer to the agent who receives the commitment. Castelfranchi proposes that commitments are established in the presence of a *witness*, who as it were officiates at the event [Castelfranchi, 1993]. We generalize this idea to the *context* of a commitment. Intuitively, the context includes the norms or conventions that apply in the given society as well as the situation at the time the commitment is instantiated. Formally, we treat the context as a group that contains the participating agents.

3.2 Discharge and Cancelation

An agent's commitments constrain him. Typically, the agent acts according to his commitments, and succeeds in satisfying them. An agent's commitments are discharged when the desired condition is obtained. This condition might be a plain physical requirement, or a requirement that some other agent recognized that some physical requirement is satisfied. Although this is the normal case, it is by no means the only case. In most settings where agents and multiagent systems are useful, inflexibility is undesirable and the agents must retain some autonomy. Commitments that cannot be discharged any more, or which the commiter does not wish to discharge, must be *canceled*. In real life, almost any commitment can be canceled. Obviously, commitments should not be canceled arbitrarily, because that would subvert their very purpose.

For this reason, we require that cancelation occur in the context group of the original commitment. Intuitively, the higher group, which contains all parties to a commitment, can absolve them of the responsibility of carrying out that commitment. Although conceptually cancelation happens in the context group, the context group may not be explicitly involved when some *cancelation policies* apply.

4 Formalizing Spheres of Commitment

We now show how commitments can be formalized. For reasons of space, we concentrate on the key intuitions, and do not give a formal syntax and semantics for our approach. We also ignore the temporal aspects.

We postulate two kinds of agents: named individuals and named groups. Individuals are considered unstructured primitives; groups are constructed from individuals or other groups, by specifying a social structure. Below we give some hints about how the structure is to be specified. By naming agents, especially groups, we can allow groups to change while maintaining a constant identity. This enables us to talk of the commitments relating to such agents.

For reasons that will become clear shortly, we postulate that commitments themselves are named objects. A commitment links the commiter, the commitee, the context, and the discharge condition. The discharge condition can be about the state of the environment, beliefs and intentions of the various agents, or about additional commitments. A commitment is satisfied when its discharge condition is met.

4.1 Social Actions

We distinguish two main kinds of actions: physical and social. Our focus here is on social actions, which are essentially communications as they affect the commitments of agents. In our framework, the following operations can be performed on commitments.

- *create*
- (satisfactorily) *discharge*
- *cancel*

When instantiated on a specific instance of a commitment, each of the above yields a social action. In the case of discharge, for simplicity, we postulate a social action in conjunction with any physical actions that may be necessary. The discharge and cancelation actions are typically not performed by the commiter, but by some other agent (the physical actions that lead to discharge would typically be performed or caused by the commiter). The agent who performs the discharges a commitment could be the commitee or the context group, who judges whether the commitment was successfully fulfilled.

4.2 Logical Form

Our correctness condition can be stated as *if a commitment is created, then it must satisfactorily discharged, unless it is canceled in time.* Cancelation can take the form of an explicit *release* by the context group, which recall, is defined to include both the commiter and the commitee. However, it is cumbersome to involve the context group in the functioning of the member agents—this is like going to court for every business deal. Instead, we propose the establishment of *cancelation policies*, under which the commiter can be released from the given commitment without involving the context group. The above lead to the following logical form for commitments:

- $\mathsf{C}(x, y, p, G, d)$, where x is the commiter, y the commitee, G the context group, p the discharge condition, and d the cancelation policy (formally a proposition).

When d comes to hold, the corresponding commitment can be canceled, but does not have to be canceled. If appropriate, d can encode the penalties to be levied on the commiter for canceling. For example, if someone breaks a deal to buy a house, they would forfeit the deposit.

Explicitly naming the commitment itself enables setting up mutual commitments. For example, we might have mutual commitments between x and y as follows: $c_1 = \mathsf{C}(x, y, p_1, G, d_1)$ and $c_2 = \mathsf{C}(x, y, p_2, G, d_2)$, where $d_1 = \mathit{cancel}(c_2)$ and $d_2 = \mathit{cancel}(c_1)$. Thus, commitment c_1 can be canceled only if c_2 is canceled, and vice versa. As stated, the two commitments can be canceled only if the cancelations happen simultaneously, or if we tolerate a temporary violation of the requirements.

We now discuss some possible cancelation policies, which relate to different situations (let the given commitment be $c = \mathsf{C}(x, y, p, G, d)$):

1. $d = \mathsf{false}$: the commitment is irrevocable. The traditional situation of section 2.1 can be modeled as $c_1 = \mathsf{C}(x, y, p_1, G, d_1)$ and $c_2 = \mathsf{C}(x, y, p_2, G, d_2)$, where d_1 and d_2 are both false.

2. $d = \mathsf{true}$: the commitment can be given up at will, and is effectively not a commitment at all.

3. $d = \mathit{cancel}(c')$, for some other commitment c'. This can be used to set up mutual commit dependencies.

4. $d = \mathit{release}(y, c)$: when the commitee must explicitly release the commiter.

5. $d = \mathit{release}(G, c)$: when the context must explicitly release the commiter.

With respect to discharge, we attach the *release* requirements into the main proposition of a commitment. For example, it is possible to commit to "making the sky green," or "making the sky appear green to the commitee" (these are different commitments, with different chances of satisfiability). Commitments can also apply to the creation, discharge, or cancelation of other commitments. This enables commitment and cancelation policies to be set up, which enable a commitment to be created or canceled under appropriate conditions.

Although we have a rich language for talking about commitments, certain logical properties can be derived. Assuming entailment of propositions, we define entailment of commitments as $c_i \rightarrow c_j \triangleq (c_i.p \rightarrow c_j.p \& c_i.d \rightarrow c_j.d)$. The motivation for this is that c_i is satisfied by any computation on which either p or d occurs; the above definition ensures that c_j is satisfied whenever c_i is satisfied. Two commitments are equivalent if they entail each other: $c_i \equiv c_j \triangleq (c_i \rightarrow c_j \& c_j \rightarrow c_i)$. We can establish some stronger results—e.g., given $c_1 = \mathsf{C}(x, y, p_1, G, d_1)$ and $c_2 = \mathsf{C}(x, y, p_2, G, d_2)$, if $p_1 \equiv \neg p_2$, then at least one of c_1 and c_2 must be canceled. Thus, for consistency, $d_1 \vee d_2$ must (eventually) hold.

However, a formal language does not guarantee that every sentence expressed in it will be intuitively meaningful! It is obviously possible, as we showed above, to capture the traditional inflexible form of commitment. The key benefit is that richer forms of commitment can also be captured in a uniform declarative manner. Because commitments are named, it is possible to have commitments that are self-referential, and whose semantics is not clear.

4.3 Social Policies

Social policies are policies that govern the social actions—they characterize when the associated action occurs. It is helpful to define the *order* of a commitment as follows. Consider a commitment $c = \mathsf{C}(x, y, p, G, d)$. c is 0-order iff p makes no reference to any commitments. c is $(i+1)$-order iff the highest order commitment referred to in p is i-order. Social policies are formally represented as conditional expressions. Policies for i-order commitments are $(i+1)$-order commitments.

For example, an agent, x, might adopt a policy of being altruistic, which might be realized as adopting commitments based on requests from others. Thus we would have $(\forall y, G, p : \mathsf{C}(x, y, \mathrm{request}(y, x, p) \rightarrow \mathit{create}(\mathsf{C}(x, y, p, G, \mathit{release}(y, \mathit{this}))), G, \mathsf{false}))$. Now when some agent y requests that x bring about or perform p, x adopts a commitment for p. Here *this* refers to the commitment itself. Additional policies can be similarly captured in our framework.

Social structure and social policies are two faces of the same coin. The former applies within a group; the latter apply across agents, including those who constitute a group. The policies of agents essentially characterize the structure of their groups. Thus if an agent x always adopts the requests of a specific agent y in a specific group G, that might represent that x is subordinate to y in G—this gives a description of the structure of G.

Policies also have a computational significance, which is that they can lead to commitments being created, discharged, or canceled without reference to the context group. It is their locality that makes policies useful in practice. Let us now exercise the above definitions on a simple example.

Example 4 Consider a number of faculty agents who agree to meet for a student's exam. The context group is the department. If all of them show up, they can discharge their commitments. Suppose one of them has to reschedule. He must first *cancel* from the previous schedule. He can do this by invoking the context group, which might have a designated deciding agent (the head). On the other hand, if the context group provides the ground rules for rescheduling exams, those ground rules can be invoked directly. For example, the rule might be to seek every other attendee's permission and to find an acceptable slot within 2 weeks of the original. If the rescheduling agent can indeed do this, he does not need to invoke the context explicitly. If he thinks the ground rules can be challenged, because it is spring break, he must invoke the context explicitly, and attempt to change the rules. ∎

Sometimes, however, it is useful to perform actions at the level of the context group. These might be necessary when the actions of the member agents need to be atomically performed or undone.

Example 5 Suppose a trip is planned that involves an airline, a hotel, and car rental company. If this plan must be rescheduled because the flight got canceled, the travel agency might prefer to cancel the three subactions and redo each of them for the new schedule—e.g., to use a hotel that has a later arrival time, and a car company that gives discounts to passengers of a particular airline. In this case, the travel agency is the designated maintainer of the context. ▮

5 Comparisons and Conclusions

We sought to present the unifying principles behind commitment for single-agent and multiagent systems. We showed how commitments relate to various aspects of agent and system architecture, and the roles they might play in multiagent systems. We are studying the database-specific ramifications of our approach.

An interesting consequence of our approach is that it relies purely on social concepts to formalize commitments. Some previous approaches, e.g., [Levesque *et al.*, 1990; Grosz & Sidner, 1990], postulate mutual beliefs among the committing agents. Mutual beliefs involve the agents to hold beliefs about each other to unbounded levels of nesting. The disadvantages of mutual beliefs are considered in [Singh, 1996]. Mutual beliefs cannot be implemented except through additional simplifying assumptions, which is why the direct approach of using social constructs is more appealing. We conjecture that named commitments, which are reminiscent of contract numbers in business dealings, provide the necessary connections among the agents. Besides its reliance on mutual beliefs, the approach of [Levesque *et al.*, 1990] hardwires a specific approach to canceling commitments (for joint intentions)—the participating agents must achieve a mutual belief that the given commitment is off.

The approach of [Jennings, 1993] is the closest in spirit to the present approach. Jennings postulates *conventions* as ways in which to reason about commitments. Thus, he can easily generalize on [Levesque *et al.*, 1990]. However, for teams, he requires a "minimum" convention, which recalls the approach of [Levesque *et al.*, 1990]. Jennings also requires a mental state as concomitant with a joint commitment. A related system was developed that reasoned about the possibilities for cooperation before engaging in planning or action [Wittig, 1992].

While we share many of the intuitions and motivations of [Jennings, 1993] (including applications involving heterogeneous information systems), we attain greater generality through the explicit use of the structure of multiagent systems. The agents always commit in the context of their multiagent system, and sometimes to that system. This has the pleasant effect that social concepts are not made dependent on psychological concepts. The multiagent system serves as the default repository for the cancelation and commitment policies, although these can, in several useful cases, be assigned to the member agents. We believe the relationship of our approach to open nested transaction models and workflows will lead to superior multiagent systems for information applications.

Gasser describes some of the sociological issues underlying multiagent systems [Gasser, 1991]. His notion of the multiple simultaneous roles played by social agents inspired part of our discussion above. Castelfranchi studies concepts similar to those here [Castelfranchi, 1993]. However, he distinguishes a notion of collective commitment, which is subsumed by our concept of commitment through the orthogonal representation of the structure of multiagent systems. Tuomela develops an interesting theory of joint action and intention that bears similarities to collective commitments [Tuomela, 1991]. [Sichman *et al.*, 1994] develop a theory and interpreter for agents who can perform social reasoning. Their agents represent knowledge about one another to determine their relative autonomy or dependence for various goals. Dependence leads to joint plans for achieving the intended goals. This theory does not talk about commitments *per se*, so it is complementary to our approach.

We believe that the approach of [Wittig, 1992] can be merged with our approach to yield more sophisticated reasoning involving context, especially organizational context, and more predictable behavior when commitments must be canceled. We also believe that we will be able to provide stronger database-specific underpinnings. We also believe that our approach with its emphasis on structure and context can be married with that of [Sichman *et al.*, 1994] to lead to more sophisticated forms of social reasoning.

6 Future Work

Multiagent systems provide greater flexibility than the database approaches. However, traditional agent and multiagent approaches have not developed the level of robustness of the database approaches. The database approaches prove too rigid to apply in open, heterogeneous environments. In summary, our goals are to enhance results from advanced transaction models and workflow management to yield flexible, customizable versions of recoverability, atomicity, and durability that are suitable for multiagent systems. We believe the present paper has many of the key ideas, but much work remains.

Specific technical directions include introducing temporal aspects into the language, and relating the development of commitments to decision theoretic analyses of rational behavior.

Acknowledgments

I have benefited from discussions with several people over the years, most notably Nicholas Asher, Cristiano Castelfranchi, Les Gasser, Michael Georgeff, and Michael Huhns. I also benefited from comments by the anonymous reviewers. I alone am responsible for the contents, however.

References

[Bukhres & Elmagarmid, 1996] Bukhres, Omran A. and Elmagarmid, Ahmed K., editors. *Object-Oriented Multidatabase Systems: A Solution for Advanced Applications.* Prentice Hall.

[Castelfranchi, 1993] Castelfranchi, Cristiano; 1993. Commitments: From individual intentions to groups and organizations. In *Proceedings of the AAAI-93 Workshop on AI and Theories of Groups and Organizations: Conceptual and Empirical Research.*

[Davies, 1978] Davies, Charles T. Jr.; 1978. Data processing spheres of control. *IBM Systems Journal* 17(2):179–198.

[Elmagarmid, 1992] Elmagarmid, Ahmed K., editor. *Database Transaction Models for Advanced Applications.* Morgan Kaufmann.

[Gasser, 1991] Gasser, Les; 1991. Social conceptions of knowledge and action: DAI foundations and open systems semantics. *Artificial Intelligence* 47:107–138.

[Gray & Reuter, 1993] Gray, Jim and Reuter, Andreas; 1993. *Transaction Processing: Concepts and Techniques.* Morgan Kaufmann.

[Grosz & Sidner, 1990] Grosz, Barbara and Sidner, Candace; 1990. Plans for discourse. In Cohen, P.; Morgan, J.; and Pollack, M., editors, *SDF Benchmark Series: Intentions in Communication.* MIT Press, Cambridge, MA.

[Jennings, 1993] Jennings, N. R.; 1993. Commitments and conventions: The foundation of coordination in multi-agent systems. *The Knowledge Engineering Review* 8(3):223–250.

[Levesque *et al.*, 1990] Levesque, H. J.; Cohen, P. R.; and Nunes, J. T.; 1990. On acting together. In *AAAI.*

[Papazoglou *et al.*, 1992] Papazoglou, Mike P.; Laufmann, Steven C.; and Sellis, Timothy K.; 1992. An organizational framework for cooperating intelligent information systems. *International Journal on Intelligent and Cooperative Information Systems* 1(1):169–202.

[Sichman *et al.*, 1994] Sichman, Jaime Simão; Conte, Rosaria; Demazeau, Yves; and Castelfranchi, Cristiano; 1994. A social reasoning mechanism based on dependence networks. In *Proceedings of the 11th European Conference on Artificial Intelligence.*

[Singh & Huhns, 1994] Singh, Munindar P. and Huhns, Michael N.; 1994. Automating workflows for service provisioning: Integrating AI and database technologies. *IEEE Expert* 9(5). Special issue on *The Best of CAIA '94* with selected papers from Proceedings of the 10th IEEE Conference on Artificial Intelligence for Applications, March 1994.

[Singh, 1991] Singh, Munindar P.; 1991. Group ability and structure. In Demazeau, Y. and Müller, J.-P., editors, *Decentralized Artificial Intelligence, Volume 2.* Elsevier Science Publishers B.V. / North-Holland, Amsterdam, Holland. 127–145.

[Singh, 1996] Singh, Munindar P.; 1996. A conceptual analysis of commitments in multiagent systems. Technical Report TR-96-09, Department of Computer Science, North Carolina State University, Raleigh, NC. Available at `http://www4.ncsu.edu/eos/info/ dblab/www/ mpsingh/ papers/ mas/ commit.ps`.

[Tuomela, 1991] Tuomela, Raimo; 1991. We will do it. an analysis of group-intentions. *Philosophy and Phenomenological Research* LI(2):249–277.

[Wittig, 1992] Wittig, Thies, editor. *ARCHON: An Architecture for Multi-agent Systems.* Ellis Horwood Limited, West Sussex, UK.