# Five Guidelines for Normative Multiagent Systems

Guido BOELLA [a], Gabriella PIGOZZI [b] and Leendert VAN DER TORRE [b]
[a] *Department of Computer Science, University of Torino*
[b] *Computer Science and Communication, University of Luxembourg*

**Abstract.** In this paper we introduce and discuss five guidelines for the use of normative systems in computer science. We adopt a multiagent systems perspective, because norms are used to coordinate, organize, guide, regulate or control interaction among distributed autonomous systems. They are derived from the computer science literature. From the so-called 'normchange' definition of the first workshop on normative multiagent systems in 2005 we derive the guidelines to motivate which definition of normative multiagent system is used, to make explicit why norms are a kind of (soft) constraints deserving special analysis, and to explain why and how norms can be changed at runtime. From the so-called 'mechanism design' definition of the second workshop on normative multiagent systems in 2007 we derive the guidelines to discuss the use and role of norms as a mechanism in a game-theoretic setting, and to clarify the role of norms in the multiagent system.

**Keywords.** normative systems, deontic logic, normative multiagent systems

## 1. Introduction

Normative systems are "systems in the behavior of which norms play a role and which need normative concepts in order to be described or specified" [21, preface]. There is an increasing interest in normative systems in the computer science community, due to the observation five years ago in the so-called AgentLink Roadmap [19, Fig. 7.1], a consensus document on the future of multiagent systems research, that norms must be introduced in agent technology in the medium term (i.e., now!) for infrastructure for open communities, reasoning in open environments and trust and reputation. However, there is no consensus yet in the emerging research area of normative multiagent systems on the kind of norms to be used, or the way to use them. Consider the following lines taken from a paper review report. A norm like "You should empty your plate" may be criticized, because it is not a (generic) norm but an obligation, or a sentence not presented as a norm, such as an imperative or command like "Empty your plate!", may be criticized because it is a norm. Alternatively, a proposed normative multiagent systems may be criticized by a reviewer, because, for example, norms cannot be violated, norms cannot be changed, and so on. These criticisms suggest that more agreement on the use of norms and normative systems in computer science would be useful.

The research question of this paper is to give general guidelines for the use of "norms" and "normative systems" in computer science. During the past two decades normative systems have been studied in a research field called deontic logic in computer science (ΔEON), and normative multiagent systems may be seen as the research field where the traditional normative systems and ΔEON meet agent research. In these areas, the following two related challenges emerged to a common use of "norms" and "normative systems" in computer science.

**There are many distinct notions of "normative systems"** in the literature due to the use of the concept "norm" in distinct disciplines, just like there are many definitions of "agent" or "actor" due to its use across disciplines. Traditionally normative systems have been studied in philosophy, sociology, law, and ethics, and "norms" can therefore be, for example, social expectations, legal laws or linguistic imperatives or commands.

**The role of norms in computer science is changing** and solutions based on multiagent systems are increasing. The seventh ΔEON conference [17,18] in 2004 in Madeira, Portugal, had as special theme "deontic logic and multiagent systems," the eighth ΔEON conference in 2006 in Utrecht, the Netherlands, had as special focus "artificial normative systems" [13,12], and the ninth ΔEON conference [13,24] in Luxembourg in 2008 was co-located with the third workshop on normative multiagent systems NorMAS. Gradually the ΔEON research focus changes from logical relations among norms to, for example, agent decision making, and to systems in which norms are created and in which agents can play the role of legislators.

We approach this question of defining guidelines for normative multiagent system research by first considering two consensus definitions in the computer science literature of previous normative multiagent systems NorMAS workshops, from which we derive our guidelines.

## 2. Normative multiagent systems

Before we consider the 'normchange' and 'mechanism design' definition of normative multiagent systems, we start with a dictionary definition of normative systems. With 'normative' we mean 'conforming to or based on norms', as in *normative behavior* or *normative judgments*. According to the Merriam-Webster Online [20] Dictionary, other meanings of normative not considered here are 'of, relating to, or determining norms or standards', as in *normative tests*, or 'prescribing norms', as in *normative rules of ethics* or *normative grammar*. With 'norm' we mean 'a principle of right action binding upon the members of a group and serving to guide, control, or regulate proper and acceptable behavior'. Other meanings of 'norm' given by the Merriam-Webster Online Dictionary but not considered here are 'an authoritative standard or model', 'an average like a standard, typical pattern, widespread practice or rule in a group', and various definitions used in mathematics.

### *2.1. The normchange definition*

The first definition of a normative multiagent system emerged after two days of discussion at the first workshop on normative multiagent systems NorMAS held in 2005 as a symposium of the Artificial Intelligence and Simulation of Behaviour convention (AISB) in Hatfield, United Kingdom:

**The normchange definition.** "A normative multiagent system is a multiagent system together with normative systems in which agents on the one hand can decide whether to follow the explicitly represented norms, and on the other the normative systems specify how and in which extent the agents can modify the norms" [3].

The first three guidelines are derived from this definition. The first one concerns the explicit representation of norms, which has been interpreted either that norms must be explicitly represented in the system (the 'strong' interpretation) or that norms must be explicitly represented in the system specification (the 'weak' interpretation). The first guideline is to make explicit and motivate which interpretation is used, the strong one, the weak one, or none of them.

**Guideline 1** *Motivate which definition of normative multiagent system is used.*

The motivation for the strong interpretation of the explicit representation is to prevent a too general notion of norms. Any requirement can be seen as a norm the system has to comply with; but why should we do so? Calling every requirement a norm makes the concept empty and useless. The weak interpretation is used to study the following two important problems in normative multiagent systems.

**Norm compliance.** How to decide whether systems or organizations comply with relevant laws and regulations? For example, is a hospital organized according to medical regulations? Does a bank comply with Basel 2 regulations?

**Norm implementation.** How can we design a system such that it complies with a given set of norms? For example, how to design an auction such that agents cannot cooperate?

The second guideline follows from the fact that agents can decide whether to follow the norms. This part of the definition is borrowed from the $\Delta$EON tradition, whose founding fathers Meyer and Wieringa observe that "until recently in specifications of systems in computational environments the distinction between normative behavior (as it *should be*) and actual behavior (as it *is*) has been disregarded: mostly it is not possible to specify that some system behavior is non-normative (illegal) but nevertheless possible. Often illegal behavior is just ruled out by specification, although it is very important to be able to specify what should happen if such illegal but possible behaviors occurs!" [21, preface]. However, constraints are well studied and well understood concepts, so if a norm is a kind of constraint, the question immediately is raised what is special about them.

**Guideline 2** *Make explicit why your norms are a kind of (soft) constraints that deserve special analysis.*

Examples of issues which have been analyzed for norms but to a less degree for other kinds of constraints are ways to deal with violations, representation of permissive norms, the evolution of norms over time (in deontic logic), the relation between the cognitive abilities of agents and the global properties of norms, how agents can acquire norms, how agents can violate norms, how an agent can be autonomous [9] (in normative agent architectures and decision making), how norms are created by a legislator, emerge spontaneously or are negotiated among the agents, how norms are enforced, how constitutive or counts-as norms are used to describe institutions, how norms are related to other social and legal concepts, how norms structure organizations, how norms coordinate groups and societies, how contracts are related to contract frames and contract law, how legal courts are related, and how normative systems interact?

For example, the norms of global policies may be represented as soft constraints, which are used in detective control systems where violations can be detected, instead of hard constraints restricted to preventative control systems in which violations are impossible. The typical example of the former is that you can enter a train without a ticket, but you may be checked and sanctioned, and an example of the latter is that you cannot enter a metro station without a ticket. However, if the norms are represented as constraints, then how to analyze that detective control is the result of actions of agents and therefore subject to errors and influenceable by actions of other agents? For example, it may be the case that violations are not often enough detected, that law enforcement is lazy or can be bribed, there are conflicting obligations in the normative system, that agents are able to block the sanction, block the prosecution, update the normative system, etc.

The third guideline follows from the fact that norms can be changed by the agents or the system, which distinguished this definition of normative multiagent system from the common framework used in the ΔEON community, and led to the identification of this definition as the "normchange" definition of normative multiagent systems.

**Guideline 3** *Explain why and how norms can be changed at runtime.*

For example, a norm can be made by an agent, as legislators do in a legal system, or there can be an algorithm that observes agent behavior, and suggests a norm when it observes a pattern. The agents can vote on the acceptance of the norm. Likewise, if the system observes that a norm is often violated, then apparently the norm does not work as desired, and it undermines the trust of the agents in the normative system, so the system can suggest that the agents can vote whether to retract or change the norm.

### *2.2. The mechanism design definition*

The fourth and fifth guideline follow from the consensus definition of the second workshop on normative multiagent systems NorMAS held as Dagstuhl Seminar 07122 in 2007. After four days of discussion, the participants agreed to the following consensus definition:

**The mechanism design definition.** "A normative multiagent system is a multiagent system organized by means of mechanisms to represent, communicate, distribute, detect, create, modify, and enforce norms, and mechanisms to deliberate about norms and detect norm violation and fulfilment." [5]

The fourth guideline emphasizes the game-theoretic model and the notion of a norm as a mechanism. According to Boella *et al.*, "the emphasis has shifted from representation issues to the mechanisms used by agents to coordinate themselves, and in general to organize the multiagent system. Norms are communicated, for example, since agents in open systems can join a multiagent system whose norms are not known. Norms are distributed among agents, for example, since when new norms emerge the agent could find a new coalition to achieve its goals. Norm violations and norm compliance are detected, for example, since spontaneous emergence norms of among agents implies that norm enforcement cannot be delegated to the multiagent infrastructure." [5]

**Guideline 4** *Discuss the use and role of norms always as a mechanism in a game-theoretic setting.*

Here we refer to game theory in a very liberal sense, not only to classical game theory studied in economics, which has been criticized for its ideality assumptions. Of particular interest are alternatives taking the limited or bounded rationality of decision makers into account. For example, Newell [22] and others develop theories in artificial intelligence and agent theory, replace probabilities and utilities by informational (knowledge, belief) and motivational attitudes (goal, desire), and the decision rule by a process of deliberation. Bratman [6] further extends such theories with intentions for sequential decisions and norms for multiagent decision making. Alternatively, Gmytrasiewitcz and Durfee [10] replace the equilibria analysis in game theory by recursive modelling, which considers the practical limitations of agents in realistic settings such as acquiring knowledge and reasoning so that an agent can build only a finite nesting of models about other agents' decisions.

Games can explain that norms should satisfy various properties to be effective as a mechanism to obtain desirable behavior. For example, the system should not sanction without reason, as for example Caligula or Nero did in the ancient Roman times, as the norms would loose their force to motivate agents. Moreover, sanctions should not be too low, but they also should not be too high, as shown by argument of Beccaria. Otherwise, once a norm is violated, there is no way to prevent further norm violations.

Games can explain also the role of various kinds of norms in a system. For example, assume that norms are added to the system one after the other and this operation is performed by different authorities at different levels of the hierarchy. Lewis "master and slave" game [16] shows that the notion of permission alone is not enough to build a normative system, because only obligations divide the possible actions into two categories or spheres: the sphere of prohibited actions and the sphere of permitted (i.e., not forbidden) actions or "the sphere of permissibility". More importantly, Bulygin [7] explains why permissive norms are needed in normative systems using his "Rex, Minister and Subject" game. "Suppose that Rex, tired of governing alone, decides one day to appoint a Minister and to endow him with legislative power. [...] an action commanded by Minister becomes as obligatory as if it would have been commanded by Rex. But Minister has no competence to alter the commands and permissions given by Rex." If Rex permits hunting on Saturday and then Minister prohibits it for the whole week, its prohibition on Saturday remains with no effect.

As another example, in our game theoretic approach to normative systems [4] we study the following kind of normative games.

**Violation games:** interacting with normative systems, obligation mechanism, with applications in trust, fraud and deception.

**Institutionalized games:** counts-as mechanism, with applications in distributed systems, grid, p2p, virtual communities.

**Negotiation games:** MAS interaction in a normative system, norm creation action mechanism, with applications in electronic commerce and contracting.

**Norm creation games:** multiagent system structure of a normative system, permission mechanism, with applications in legal theory.

**Control games:** interaction among normative systems, nested norms mechanism, with applications in security and secure knowledge management systems.

The fifth guideline follows from the introduction of organizational issues in the definition of normative multiagent systems. Norms are no longer seen as the mechanism to regulate behavior of the system, but part of a larger institution. This raises the question what precisely the role of norms is in such an organization.

**Guideline 5** *Clarify the role of norms in your system.*

Norms are rules used to guide, control, or regulate desired system behavior. However, this is not unproblematic. For example, consider solving traffic problems by introducing norms, as a cheap alternative to building new roads. It does not work, for the following two reasons. The first reason is that if you change the system by building new norms or introducing new norms, then people will adjust their behavior. For example, when roads improve, people tend to live further away from their work. In other words, a normative multiagent system is a self-organizing system. Moreover, the second problem with norm design is that norms can be violated. For example, most traffic is short distance, for which we could forbid using the car. However, it is hard to enforce such a norm, since people will always claim to have come from long distance, even if they live around the corner.

Norms can also be seen as one of the possible incentives to motivate agents, which brings us again back to economics.

> "Economics is, at root, the study of incentives: how people get what they want, or need, especially when other people want or need the same thing. Economists love incentives. They love to dream them up and enact them, study them and tinker with them. The typical economist believes the world has not yet invented a problem that he cannot fix if given a free hand to design the proper incentive scheme. His solution may not always be pretty–but the original problem, rest assured, will be fixed. An incentive is a bullet, a lever, a key: an often tiny object with astonishing power to change a situation.
> . . .
> There are three basic flavors of incentive: economic, social, and moral. Very often a single incentive scheme will include all three varieties. Think about the anti-smoking campaign of recent years. The addition of $3-per-pack "sin tax" is a strong economic incentive against buying cigarettes. The banning of cigarettes in restaurants and bars is a powerful social incentive. And when the U.S. government asserts that terrorists raise money by selling black-market cigarettes, that acts as a rather jarring moral incentive.' [15]

Here it is important to see that moral incentives are very different from financial incentives. For example, Levitt [15, p.18-20], discussing an example of Gneezy and Rustichini [11], explains that the number of violations may *increase* when financial sanctions are imposed, because the moral incentive to comply with the norm is destroyed. The fact that norms can be used as a mechanism to obtain desirable system behavior, i.e. that norms can be used as incentives for agents, implies that in some circumstances economic incentives are not sufficient to obtain such behavior. For example, in a widely discussed example of the so-called centipede game, there is a pile of thousand pennies, and two agents can in turn either take one or two pennies. If an agent takes one then the other agent takes turn, if it takes two then the game ends. A backward induction argument implies that it is rational only to take two at the first turn. Norms and trust have been discussed to analyze this behavior, see [14] for a discussion.

A rather different role of norms is to organize systems. To manage properly complex systems like multiagent systems, it is necessary that they have a modular design. While in traditional software systems, modularity is addressed via the notions of class and object, in multiagent systems the notion of organization is borrowed from the ontology of social systems. Organizing a multiagent system allows to decompose it and defining different levels of abstraction when designing it. Norms are another answer to the question of how to model organizations as first class citizens in multiagent systems. Norms are not usually addressed to individual agents, but rather they are addressed to roles played by agents [2]. In this way, norms from a mechanism to obtain the behavior of agents, also become a mechanism to create the organizational structure of multiagent systems. The aim of an organizational structure is to coordinate the behavior of agents so to perform complex tasks which cannot be done by individual agents. In organizing a system all types of norms are necessary, in particular, constitutive norms, which are used to assign powers to agents playing roles inside the organization. Such powers allow to give commands to other agents, make formal communications and to restructure the organization itself, for example, by managing the assignment of agents to roles. Moreover, normative systems allow to model also the structure of an organization and not only the interdependencies among the agents of an organization. Consider a simple example from organizational theory in Economics: an enterprise which is composed by a direction area and a production area. The direction area is composed by the CEO and the board. The board is composed by a set of administrators. The production area is composed by two production units; each production unit by a set of workers. The direction area, the board, the production area and the production units are functional areas. In particular, the direction area and the production areas belong to the organization, the board to the direction area, etc. The CEO, the administrators and the members of the production units are roles, each one belonging to a functional area, e.g., the CEO is part of the direction area. This recursive decomposition terminates with roles: roles, unlike organizations and functional areas, are not composed by further social entities. Rather, roles are played by other agents, real agents (human or software) who have to act as expected by their role. Each of these elements can be seen as an institution in a normative system, where legal institutions are defined by Ruiter [23] as "systems of [regulative and constitutive] rules that provide frameworks for social action within larger rule-governed settings". They are "relatively independent institutional legal orders within the comprehensive legal orders".

## 3. Summary

Next generation normative multiagent systems contain general and domain independent norms by combining three existing representations of normative multiagent systems.

First, theories of normative systems and deontic logic, the logic of obligations and permissions, for the explicit representation of norms as rules, the application of such rules, contrary-to-duty reasoning and the relation to permissions. Second, agent architecture for software engineering of agents and a model of normative decision making. Third, a game-theoretic approach for model of interaction explaining the relation among social norms and obligations, relating regulative norms to constitutive norms, the evolution of normative systems, and much more. In this paper, we introduce and discuss five guidelines for the development of normative multiagent systems. In an extended version of the present paper [1], we extend the list with five guidelines inspired by the philosophical literature.

1. Motivate which definition of normative multiagent system is used.
2. Make explicit why norms are a kind of (soft) constraints deserving special analysis.
3. Explain why and how norms can be changed at runtime.
4. Discuss the use and role of norms as a mechanism in a game-theoretic setting.
5. Clarify the role of norms in the multiagent system.
6. Relate the notion of "norm" to the legal, social, or moral literature.
7. Use norms not only to distinguish right from wrong, but also to resolve dilemmas, and use norms not only describe violations, but in general to coordinate, organize, guide, regulate or control interaction among agents.
8. Distinguish norms from obligations, prohibitions and permissions.
9. Use the deontic paradoxes only to illustrate the normative multiagent system.
10. Consider regulative norms in relation to other kinds of norms and concepts.

**Table 1.** Ten guidelines for the development of normative multiagent systems

The use of norms and normative systems in computer science are examples of the use of social concepts in computer science, which is now so well-established that the original meaning of some of these concepts in the social sciences is sometimes forgotten. For example, the original meaning of a "service" in business economics is rarely considered by computer scientists working on service oriented architectures or web services, and likewise for service level agreements and contracts, or quality of service. some social concepts have various new meanings. For example, before its use in service level agreements, the notion of "contract" was introduced in software engineering in Meyer's design by contract, a well known software design methodology that views software construction as based on contracts between clients (callers) and suppliers (routines), assertions, that has been developed in the context of object oriented and the basis of the programming language Eiffel. "Coordination" is emerging as an interdisciplinary concept to deal with the complexity of compositionality and interaction, and has been used from coordination languages in software engineering to a general interaction concept in multiagent systems. In the context of information security and access control "roles" became popular, with the popularity of eBay, the social concepts of "trust" and "reputation" have become popular, and with the emergence of social interaction sites like FaceBook or Second Life, new social concepts like societies, coalitions, organizations, institutions, norms, power, and trust are emerging [8]. In multiagent systems, social ability as the interaction with other agents and co-operation is one of the three meanings of flexibility in flexible autonomous action in Wooldridge and Jennings' weak notion of agency [25]; the other two are reactivity as interaction with the environment, and pro-activeness as taking the initiative.

The main open question is whether “norms” could (or should) play a similar role in computer science like “service”, “contract” or “trust”? One suggestion comes from human computer interaction. Since the use of norms is a key element of human social intelligence, norms may be essential too for artificial agents that collaborate with humans, or that are to display behavior comparable to human intelligent behavior. By integrating norms and individual intelligence normative multiagent systems provide a promising model for human and artificial agent cooperation and co-ordination, group decision making, multiagent organizations, regulated societies, electronic institutions, secure multiagent systems, and so on. Another suggestion comes from the shared interest of multiagent system research and sociology in the relation between micro-level agent behaviour and macro-level system effects. Norms are thought to ensure efficiency at the level of the multiagent system whilst respecting individual autonomy. However, all these and other suggestions bring circumstantial evidence at best. We have to build more flexible normative multiagent systems, and test them in practice, before we know where they can be used best.

For further reading on the use of normative systems in computer science, we recommend the proceedings of the ΔEON conferences and the normative multiagent systems workshops. The abstracts of all papers that appeared at DLCS conferences can be searched on the deontic logic website:

`http:\\deonticlogic.org`

## References


[1] G. Boella, G. Pigozzi, and L. van der Torre. Norms in computer science: Ten guidelines for normative multiagent systems. In *Informal proceedings of NORMAS'09*, 2009.

[2] G. Boella and L. van der Torre. The ontological properties of social roles in multi-agent systems: Definitional dependence, powers and roles playing roles. *Artificial Intelligence and Law Journal (AILaw)*, 2007.

[3] G. Boella, L. van der Torre, and H. Verhagen. Introduction to normative multiagent systems. *Computation and Mathematical Organizational Theory, special issue on normative multiagent systems*, 12(2-3):71–79, 2006.

[4] G. Boella, L. van der Torre, and H. Verhagen. Normative multi-agent systems. In *Internationales Begegnungs und Porschungszentrum fur Informatik (IBFI)*, 2007.

[5] G. Boella, H. Verhagen, and L. van der Torre. Introduction to the special issue on normative multiagent systems. *Journal of Autonomous Agents and Multi Agent Systems*, 17(1):1–10, 2008.

[6] M.E. Bratman. *Intentions, plans, and practical reason*. Harvard University Press, Harvard (Massachusetts), 1987.

[7] E. Bulygin. Permissive norms and normative systems. In A. Martino and F. Socci Natali, editors, *Automated Analysis of Legal Texts*, pages 211–218. Publishing Company, Amsterdam, 1986.

[8] C. Castelfranchi. The micro-macro constitution of power. *Protosociology*, 18:208–269, 2003.

[9] R. Conte, C. Castelfranchi, and F. Dignum. Autonomous norm-acceptance. In *Intelligent Agents V (ATAL98)*, LNAI 1555, pages 319–333. Springer, 1999.

[10] P. J. Gmytrasiewicz and E. H. Durfee. Formalization of recursive modeling. In *Procs. of ICMAS'95*, pages 125–132, Cambridge (MA), 1995. AAAI/MIT Press.

[11] U. Gneezy and A. Rustichini. A fine is a price. *The Journal of Legal Studies*, 29(1):1–18, 2000.

[12] L. Goble and J.J. Ch. Meyer, editors. *Deontic Logic and Artificial Normative Systems, 8th International Workshop on Deontic Logic in Computer Science, DEON 2006, Utrecht, The Netherlands, July 12-14, 2006, Proceedings*, volume 4048 of *Lecture Notes in Computer Science*. Springer, 2006.

[13] L. Goble and J.J. Ch. Meyer. Revised versions of papers presented in the proceeding of the eighth international workshop on deontic logic in computer science (DEON06). *Journal of Applied Logic*, 6(2), 2008.

[14] M. Hollis. *Trust within reason*. Cambridge University Press, Cambridge, 1998.

[15] Steven D. Levitt and Stephen J. Dubner. *Freakonomics : A Rogue Economist Explores the Hidden Side of Everything*. William Morrow, New York, May 2005.

[16] D. Lewis. A problem about permission. In E. Saarinen, editor, *Essays in Honour of Jaakko Hintikka*, pages 163–175. D. Reidel, Dordrecht, 1979.

[17] A. Lomuscio and D. Nute, editors. *Deontic Logic in Computer Science, 7th International Workshop on Deontic Logic in Computer Science, DEON 2004, Madeira, Portugal, May 26-28, 2004. Proceedings*, volume 3065 of *Lecture Notes in Computer Science*. Springer, 2004.

[18] A. Lomuscio and D. Nute. Revised versions of papers presented in the proceeding of the seventh international workshop on deontic logic in computer science (DEON04). *Journal of Applied Logic*, 3(3-4), 2005.

[19] M. Luck, P. McBurney, and C. Preist. *Agent Technology: Enabling Next Generation Computing (A Roadmap for Agent Based Computing)*. AgentLink, 2003.

[20] Merriam-Webster. *Online dictionary http://www.merriam-webster.com/*. Merriam-Webster.

[21] J.-J. Meyer and R. Wieringa. *Deontic Logic in Computer Science: Normative System Specification*. John Wiley & Sons, Chichester, England, 1993.

[22] A. Newell. The knowledge level. *Artificial Intelligence*, 18:87–127, 1982.

[23] D.W.P. Ruiter. A basic classification of legal institutions. *Ratio Juris*, 10(4):357–371, 1997.

[24] R. van der Meyden and L. van der Torre, editors. *Deontic Logic in Computer Science, 9th International Conference on Deontic Logic in Computer Science, DEON 2008, Luxembourg, July 16-18, 2008, Proceedings*, LNCS, Berlin, in press. Springer.

[25] M. J. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10(2):115–152, 1995.