Coordination in Normative Multiagent Systems

Guido Boella [1]

Dipartimento di Informatica
Università di Torino
Torino, Italy

Joris Hulstijn [2]

Department
Vrije Universiteit
Amsterdam, The Netherlands

Leendert van der Torre [3]

CWI Amsterdam
Delft University of Technology
The Netherlands

Abstract

We survey some of our work in deontic logic and normative multiagent systems. First, we discuss how deontic logic may help the coordination of components by discussing the use of logic as a secretarial assistant, which facilitates exogenous coordination. Secondly, we discuss which concepts normative multiagent systems to offer to coordination, by discussing contracts and roles. Thirdly, we discuss how social commitments help coordination, by discussing the distinction and possible synthesis between the FIPA approach and the social semantics approach.

Key words: Normative systems, multiagent systems, deontic logic, coordination, communication.

1 Introduction

Norms play an important role in the coordination of social systems. Normative multiagent systems are "sets of agents (human or artificial) whose interactions

[1] Email: guido@di.unito.it
[2] Email: jhulstijn@feweb.vu.nl
[3] Email: torre@cwi

This is a preliminary version. The final version will be published in
Electronic Notes in Theoretical Computer Science
URL: www.elsevier.nl/locate/entcs

can fruitfully be regarded as norm-governed; the norms prescribe how the agents ideally should and should not behave" [10]. Moreover, norms have also been studied in deontic logic, a branch of philosophical logic that studies logical relations among obligations, permissions and prohibitions. Deontic logic has been developed to model legal and moral systems, though it has also been applied to problems in computer science that involve soft constraints, i.e., constraints that can be violated [17].

Coordination has been studied in many other areas too. According to many in computer science, the interaction paradigm provides a new conceptualization of computational phenomena that emphasize interaction rather than algorithms: concurrent, distributed, reactive, embedded, component-oriented, agent-oriented and service-oriented systems all exploit interaction as a fundamental paradigm [2].

In this paper we raise the question what formal approaches to normative systems have to offer to the coordination community. We use our own work to illustrate the use of deontic logic, concepts in normative multiagent systems, and social commitments in communication. Concerning logic, we discuss the use of logic as secretarial assistant in a deontic logic called input/output logic. Concerning concepts, we discuss the use of contracts and roles. Concerning communication, we discuss the role of social commitments. The layout of this paper follows these three issues.

2 Logic

Makinson and van der Torre [14,11] argue that the new role of logic in agent theory is not to study some kind of non-classical logic, but a way of using the classical one. From a very general perspective, logic is often seen as an inference motor, with premises as inputs and conclusions as outputs. But it may also be seen in another role, as secretarial assistant to some other, perhaps non-logical, transformation engine. From this point of view, the task of logic is one of preparing inputs before they go into the machine, unpacking outputs as they emerge and, less obviously, *co-ordinating* the two. The process as a whole is one of logically assisted transformation, and is an inference only when the central transformation is so. This is the general perspective underlying input/output logics. It is one of logic at work rather than logic in isolation.

This secretarial role of logic is inherently amenable to exogenous coordination. For example, compare it to the notion of abstract behavior type (ABT), which has been proposed by Arbab [1] as a proper foundation model of coordination - more precisely, as a higher-level alternative to abstract data type (ADT). The ABT model supports a much looser coupling than is possible with the ADT's operational interface. The point is that ABTs do not specify any detail about the operations that may be used to implement such behavior or the data types it may manipulate for its realization - just like input/output logic does not specify the black box.

2.1 Unconstrained Input/Output Operations

Imagine a black box into which we may feed propositions as input, and that also produces propositions as output. Of course, classical consequence may itself be seen in this way, but it is a very special case, with additional features - inputs are also themselves outputs, since any proposition classically implies itself, and the operation is in a certain sense reversible, since contraposition is valid. However, there are many examples without those features. In the two kinds of examples distinguished in [11], the outputs either express some kind of belief or expectation, or what is deemed desirable in conditions given by the inputs.

Technically, a normative code is seen as a set G of conditional norms, i.e. a set of such ordered pairs (a, x). For each such pair, the body a is thought of as an input, representing some condition or situation, and the head x is thought of as an output, representing what the norm tells us to be desirable, obligatory or whatever in that situation. The task of logic is seen as a modest one. It is not to create or determine a distinguished set of norms, but rather to prepare information before it goes in as input to such a set G, to unpack output as it emerges and, if needed, coordinate the two in certain ways. A set G of conditional norms is thus seen as a transformation device, and the task of logic is to act as its secretarial assistant.

In the simplest kind of unconstrained input/output operations, a set A of propositions serves as explicit input, which is prepared by being expanded to its classical closure $Cn(A)$. This is then passed into the black box or transformer G, which delivers the corresponding immediate output $G(Cn(A)) = \{x : \exists a \in Cn(A), (a, x) \in G\}$. Finally, this is expanded by classical closure again into the full output $out_1(G, A) = Cn(G(Cn(A)))$. We call this simple-minded output. This is already an interesting operation. As desired, it does not satisfy the principle of identity, which in this context we call throughput, i.e. in general we do not have $a \in out_1(G, a)$ which we write briefly, dropping the parentheses, as $out_1(G, a)$.

It is characterized by three rules. Writing $x \in out_1(G, a)$ as $(a, x) \in out_1(G)$ and dropping the right hand side as G is held constant, these rules are:

Strengthening Input (SI): From (a, x) to (b, x) whenever $a \in Cn(b)$

Conjoining Output (AND): From (a, x), (a, y) to $(a, x \wedge y)$

Weakening Output (WO): From (a, x) to (a, y) whenever $y \in Cn(x)$.

But simple-minded output lacks certain features that may be desirable in some contexts. In the first place, the preparation of inputs is not very sophisticated. Consider two inputs a and b. By classical logic, if $x \in Cn(a)$ and $x \in Cn(b)$ then $x \in Cn(a \vee b)$. But there is nothing to tell us that if $x \in out_1(G, a) = Cn(G(Cn(a)))$ and $x \in out_1(G, b) = Cn(G(Cn(b)))$ then $x \in out_1(G, a \vee b) = Cn(G(Cn(a \vee b)))$.

In the second place, even when we do not want inputs to be automatically carried through as outputs, we may still want outputs to be reusable as inputs which is quite a different matter. Operations satisfying each of these two features can be provided with explicit definitions, characterized by straightforward rules. We thus have four very natural systems of input/output, which are labelled as follows: simple-minded alias out_1 (as above), basic (simple-minded plus input disjunction: out_2), reusable (simple-minded plus reusability: out_3), and reusable basic (all together: out_4).

The three stronger systems may also be characterized by adding one or both of the following rules to those for simple-minded output:

- Disjoining input (OR): From (a, x), (b, x) to $(a.b, x)$
- Cumulative transitivity (CT): From (a,x), (a.x,y) to (a,y).

These four operations have four counterparts that also allow throughput. Intuitively, this amounts to requiring $A \subseteq G(A)$. In terms of the definitions, it is to require that G is expanded to contain the diagonal, i.e. all pairs (a, a). Derivationally, it is to allow arbitrary pairs of the form (a, a) to appear as leaves of a derivation; this is called the zero-premise identity rule ID. All eight systems are distinct, with one exception: basic throughput, which we write as out_2^+, authorizes reusability, so that $out_2^+ = out_4^+$.

2.2 Extensions

The motivation comes from the logic of conditional norms, where we need an approach that does not presume that directives carry truth-values. Input/output logic seeks to extract the essential mathematical structure behind recent attempts to reconstruct deontic logic that avoid treating norms as if they had truth-values. However, it can be used for other input/output operations with exogenous coordination as well.

Unconstrained input/output provides us with a simple and elegant construction, with straightforward behavior, but whose application to norms totally ignores the subtleties of violations and exceptions. Therefore input/output operations may be subjected to consistency constraints [12]. Moreover, input/output logics also provide a convenient platform for distinguishing and analyzing several different kinds of permission [13]. Input/output operations also enable us to give a clear formal articulation of the well-known distinction between negative and positive permission. They also enable us, for the first time, to distinguish two very different kinds of positive permission, with quite different uses in practical life.

From a coordination perspective, we should investigate two extensions. First, structured assemblies of input/output operations, called logical input/output nets, or lions for short, are graphs, with the nodes labeled by pairs (G,out) where G is a normative code and out is an input/output operations (or recursively, by other lions). The relation of the graph indicates which nodes have access to others, providing passage for the transmission of local outputs as local inputs. The graph is further equipped with an entry point and an exit point, for global input and output. Second, we have to make time explicit. For example, an abstract behavior type defines an abstract behavior as a relation among a set of timed-data-streams.

3 Concepts

Coordination in normative multiagent systems [4] uses besides norms also new concepts like contracts [6] and roles [5]. As illustration we consider normative multiagent systems built on deontic logic. In particular, there are various ways in which the input/output logics discussed thus far can be used in normative multiagent systems. On the one hand, an agent itself can be seen as a black box, with inputs the agent's observations and outputs the actions that constitute its behavior. When an abstract behavior type characterizes agent behavior, it is referred to as an agent type [8]. Moreover, in Broersen et al's BOID architectuyre [8] and in Boella and van der Torre's normative multiagent systems [4], input/output logics are used to represent mental attitudes of agents, and extended with game theoretic concepts.

3.1 Normative multiagent systems

Various kinds of norms have been studied, such as regulative norms, permissive norms, constitutive norms, social laws, etc. The Boella-van der Torre model of normative multiagent systems [4] attributed mental attitudes to normative systems, which has two important advantages:

(i) Obligations can be defined in the BDI framework. The desires or goals of the normative system are the obligations of the agent. This contributes to the open problem whether norms and obligations should be represented explicitly, for example in a deontic logic, or they can also be represented implicitly.

(ii) The interaction between an agent and the normative system can be modeled as a game between two agents. Consequently, methods and tools used in game theory such as equilibrium analysis can be applied to normative reasoning.

For example, a qualitative game theory can be developed based on recursive modeling of the normative system by the bearer of the obligation: The agent bases its decision on the consequences of the normative system's anticipated reaction, using the system's beliefs, desires and goals, in particular whether the system considers its decision as a violator and thus sanctions it. Likewise recursive games can be defined from the normative system's perspective, to for example decide which norms should be created.

Regulative norms like obligations can be defined as follows. If agent A is obliged to a, then agent N may decide that the absence of a counts as a violation of some norm n and that agent A must be sanctioned, and:

(i) Agent A believes that agent N desires that A does a.

(ii) Agent A believes that agent N desires $\neg V(n)$, that there is no violation of norm n, but if agent N believes $\neg a$ then it has the goal $V(n)$, it counts as a violation.

(iii) Agent A believes that agent N desires $\neg s$, not to sanction, but if agent N decides $V(n)$ then it has as a goal that it sanctions agent A by doing s. Agent N only sanctions in case of violation. Moreover, agent A believes that agent N has a way to apply the sanction.

(iv) Agent A desires $\neg s$: it does not like the sanction.

We can define different agent types. For example, respectful agents respect norms as such, whereas for selfish agents the only motivation to comply with obligations is the fear for sanction or the desire for reward. Intermediate agent types are possible too. A selfish agent exploits the beliefs and goals of the normative system to violate an obligation without being sanctioned.

3.2 Contracts

Coordination can be achieved without requiring a specific agent architecture, but by means of organizational design in terms of roles and norms as incentives for cooperation. The interaction structure of the organization should not be completely fixed in advance. For this purpose, some approaches introduce the possibility for agents to stipulate contracts. A contract can be defined as a statement of intent that regulates behavior among organizations and individuals.

Contracts have been proposed to make the way agents can change the interaction with and within the society explicit: they create obligations, permissions and new possibilities of interactions among agents. From a contractual perspective, organizations can be seen as the possible sets of agreements for satisfying the diverse interests of self interested individuals. We define in [6] a contract as a triple:

(i) An institutional fact $c \in I$ representing that the contract has been created.

(ii) A constitutive rule which makes true the institutional fact c

(iii) A set of constitutive rules having as antecedent the creation c of the contract and as consequent creation actions modifying the mental attitudes of the organization.

3.3 Roles

Multiagent systems are often proposed as a solution for the organizational design of open systems . A key notion in the structure of an organization is that of role. Roles allow to specify the activities delegated by the organization to achieve its overall goal while abstracting from the individuals which will eventually play them. The description of a role is usually given in terms of normative descriptions, expectations, standardised patterns of behavior, social commitments, goals and planning rules. The normative description specifies the obligations an agent who plays the role (called the actor) should obey. Goals are his intrinsic motivations. Roles, thus, seem to be strictly related to the notion of agent: they are described using notions like actions, goals and obligations.

In the Boella-van der Torre model, we attribute mental attitudes to roles and use this model for several problems. First, if we design an organization in terms of roles by attributing mental attitudes to them, we can design it like we would design an organization in terms of agents. However, the organization does not depend on the agents operating in it: they can be replaced by other ones playing the same roles. Roles roles can be delegated tasks which are reached with a greater flexibility: it would be not sufficient to describe roles as machines. Second, if we consider the problem of assigning roles to actors and we attribute mental attitudes to roles, then the problem becomes finding a match between the beliefs representing expertise of the role and the beliefs of the actor. Third, if we consider the governance of organizations, then the attribution of mental attitudes to roles makes it easier to verify whether an agent is acting according to the overall plan of the organization: if he is acting according to the role's beliefs and achieving the role's goals, then he is fulfilling his responsibilities.

The issue discussed in the formalization is which beliefs and goals are attributed to a role. Not all the beliefs of the organization are beliefs of a role due to the role assignment problem: the less beliefs a role has, the easier it becomes to assign agents to this role. Similarly for goals - not all the goals of an organization are the goals of a role due to the governance problem: to distribute responsibilities over the agents, and to control the running system. Consequently, the organizational design aims at defining roles with minimal beliefs and goals. In reality, however, the picture is more complicated than this, because not only the beliefs and goals of an organization may not be goals of the role, also vice versa a goal of a role may not be a goal of the organization.

4 Communication

In the communication of computer systems, interaction is regulated by a protocol: a set of rules that define which messages are allowed by which participants at which stage of the interaction. Well known examples are the Contract Net Protocol, that regulates a distributed procedure for allocating tasks, or protocols for negotiation. Protocols have a normative aspect: they consist of rules specifying what an agent should and shouldn't do (prescription); based on those rules other agents can form expectations about the behavior of other participants (prediction).

4.1 Speech act theory

Within the autonomous agents and multiagent systems community, agent communication standards have been developed for structuring messages and for simple interaction protocols. The backbone of these standards is formed by speech act theory [3,15]. The semantics of a speech act is commonly given by the preconditions and intended effect on the mental state of an agent, expressed using modal operators for belief and intention. Early research in AI [9] showed that the systematic analysis of speech acts in terms of preconditions and postconditions could be modeled straightforwardly in terms of planning operators. That means that a sequence of messages makes sense, in case it can be interpreted as a plan to achieve some goal.

The FIPA standardization effort has been a relative success, although it has been criticized heavily. A point of criticism is that it is impossible to verify the correct usage of a speech act, since for most realistic multiagent settings the mental state of an agent is inaccessible. Agents may well be lying. This makes it impossible to verify protocols under common assumptions regarding multiagent systems [18]. What is needed instead is a semantics that is based on *public information* about what agents are committed to, on the basis of what they have said.

4.2 Social semantics

For this reason, Singh [16] proposed a social semantics. It is based on the notion of *commitment*. Commitment binds a speaker to the community and, unlike mental attitudes, commitments have a public character. In some caases, commitments generate a kind of obligation which is endorsed by a speaker through uttering a speech act, for example: to subsequently defending the proposition, if one is challenged to do so by another speaker in dialogue.

These two approaches – mentalistic and social – are proposed as competing alternatives, but it is quite possible to combine the two approaches. The combination rests on the following key idea: if we attribute mental attitudes to the roles of the participants, rather than the individual agents that enact the roles the mental attitudes do get a public character, just like commitments.

Such a combination has as an advantage that, on the one hand applications which take place in a restricted known cooperative environment can re-use much of the work done on FIPA oompliant agent communication, but on the other hand, that stronger obligations can be added, in case the circumstances require it. In particular, in competitive or other non-cooperative circumstances, no sincerity needs to be assumed and an obligation to defend a publicly attributed belief, as as suggested above, needs to be added. However, the basic bookkeeping and the semantics of the speech acts themselves, can remain the same.

4.3 Synthesis

The method we adopt is to model dialogue as a game in which agents play roles. Speech acts are moves in the game and their preconditions and effects refer to the mental states attributed to the roles, not to the mental states of the agents themselves. A precondition of this method is that mental attitudes can be attributed to roles as well as to agents. Following [5,7], we describe roles as agents with mental attitudes, albeit of a different kind of agents, since they are not autonomous.

Agents playing roles in the game can affect the state of the game, and the state of the roles they play and of the other roles. This is possible thanks to constitutive rules which defines both the game and the roles. Constitutive rules explain how an utterance by a player counts as a move in the game and how the move impacts on the beliefs and goals of the roles. In contrast with agents' beliefs and goals, beliefs and goals of roles have a public character in that the game defines them, and it also defines how they evolve during a dialogue according to the moves played by the agents. Moreover, since roles provide a description of expected behavior they have a prescriptive character: the player of a role becomes committed to the beliefs and goals of the role created during the dialogue. Finally, the constitutive rules can describe, if necessary, how obligations are created during the dialogue.

Since games have both constitutive and regulative rules they can be considered as a normative multiagent system. Like for roles, our view of normative systems is based on the agent metaphor: a normative system can technically be considered as an agent to which mental attitudes are attributed. Moreover, normative systems, like organizations, can be articulated in roles.

4.4 Example

a	The president will win the election.	*inform*(a, b, p)
b	But there is fraud,	*inform*(b, a, q)
	so the president will not win.	*inform*$(b, a, q \supset \neg p)$
a	Fraud? But why!	*why-question*(a, b, q)
b	Fair enough, no fraud.	*retract*$(b, a,$*inform*$(b, a, q))$
	So you're right.	*accept*(b, a, p)

The effects of the constitutive rules driving the game are illustrated below, which corresponds to the example above.

Turn 1: role s informs r that p. The rules of role s now state that p is a consequence of s's beliefs. The rules of role r state that the belief p can be attributed to r unless r challenges s's inform.

Turn 2: the agent playing role r challenges s's information by means of an argument of the form $q, (q \supset \neg p)$.

Turn 3: s is facing the risk of losing the game: if it does not do anything, it will get into a contradiction, since r's argument supports $\neg p$, which is in contrast with its current beliefs (p). It has some alternatives: retracting its first *inform* or challenging r's argument. So s decides to challenge r's challenge by asking for justification (*why*).

Turn 4: r retracts the *inform* about q, thus giving up its challenge to p, and subsequently accepts p.

This example shows that indeed utterances can be modeled as count as rules, with the preconditions and postconditions of the mentalistic approach, reinterpreted as the mental attitudes of the roles.

5 Summary

In this paper we do not present any new result. Instead, this paper may be seen as a short and strongly biased introduction to normative multiagent systems.

In particular, we discuss three issues in our study of deontic logic and normative multiagent systems which may be of interest to a wider coordination community:

Logic How can deontic logic help the coordination of components? We discuss the use of logic as a secretarial assistant, which facilitates exogenous coordination.

Concepts Which concepts have normative multiagent systems to offer to coordination? We discuss contracts and roles.

Communication How do social commitments help coordination? We discussed the distinction and possible synthesis between the FIPA approach and the social semantics approach.

References

[1] Arbab, F., *Abstract behavior types: A foundation model for components and their composition*, in: *FMCO*, LNCS **2852**, Springer, 2003 pp. 33–70.

[2] Arbab, F., *Coordination of interacting concurrent computations*, in: *Interactive Computation: The New Paradigm*, Springer, 2005 .

[3] Austin, J. L., "How to Do Things with Words," Harvard University Press, Cambridge, Mass, 1962.

[4] Boella, G. and L. van der Torre, *Attributing mental attitudes to normative systems*, in: *Proceedings of the second international joint conference on autonomous agents and multi agent systems (AAMAS'03)*, 2003.

[5] Boella, G. and L. van der Torre, *Attributing mental attitudes to roles: The agent metaphor applied to organizational design*, in: *Procs. of ICEC'04* (2004).

[6] Boella, G. and L. van der Torre, *Contracts as legal institutions in organizations of autonomous agents*, in: *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS'04)*, 2004, pp. 948–955.

[7] Boella, G. and L. van der Torre, *Regulative and constitutive norms in normative multiagent systems*, in: *Procs. of 9th International Conference on the Principles of Knowledge Representation and Reasoning*, 2004, pp. 255–265.

[8] Broersen, J., M. Dastani, J. Hulstijn and L. van der Torre, *Goal generation in the BOID architecture*, Cognitive Science Quarterly **2(3-4)** (2002), pp. 428–447.

[9] Cohen, P. and H. Levesque, *Intention is choice with commitment*, Artificial Intelligence **42 (2-3)** (1990), pp. 213–261.

[10] Jones, A. and J. Carmo, *Deontic logic and contrary-to-duties*, in: D. Gabbay, editor, *Handbook of Philosophical Logic*, Kluwer, 2002 pp. 203–279.

[11] Makinson, D. and L. van der Torre, *Input-output logics*, Journal of Philosophical Logic **29** (2000), pp. 383–408.

[12] Makinson, D. and L. van der Torre, *Constraints for input-output logics*, Journal of Philosophical Logic **30** (2001), pp. 155–185.

[13] Makinson, D. and L. van der Torre, *Permissions from an input/output perspective*, Journal of Philosophical Logic **32 (4)** (2003), pp. 391–416.

[14] Makinson, D. and L. van der Torre, *What is input/output logic?*, in: *Foundations of the Formal Sciences II: Applications of Mathematical Logic in Philosophy and Linguistics*, Trends in Logic **17**, Kluwer, 2003 .

[15] Searle, J., "Speech Acts: an Essay in the Philosophy of Language," Cambridge University Press, Cambridge (UK), 1969.

[16] Singh, M. P., *A social semantics for agent communication languages*, in: F. Dignum and M. Greaves, editors, *Issues in Agent Cmmunication: Proceedings of the IJCAI Workshop on Agent Communication Languages*, LNCS 1916, Springer-Verlag, London, UK, 2000 pp. 31 – 45.

[17] Wieringa, R. and J.-J. Meyer, *Applications of deontic logic in computer science: A concise overview*, in: J.-J. Meyer and R. Wieringa, editors, *Deontic Logic in Computer Science: Normative System Specification*, John Wiley & Sons, Chichester, England, 1993 pp. 17–40.

[18] Wooldridge, M. J., *Semantic issues in the verification of agent communication languages*, Journal of Autonomous Agents and Multi-Agent Systems **3** (2000), pp. 9–31.