Formal specification of interaction in agent societies

Virginia Dignum[1,2], John-Jules Meyer[2], Frank Dignum[2], Hans Weigand[3]

[1]Achmea, The Netherlands,
email: virginia.dignum@achmea.nl

[2]University Utrecht, The Netherlands,
email: {virginia, jj, dignum}@cs.uu.nl

[3]Infolab, Tilburg University, The Netherlands,
email: weigand@uvt.nl

Abstract. The Agent Society framework that we have developed distinguishes between the mechanisms though which the structure and global behavior of the model is described and coordinated, and the aims and behavior of the service-providers (agents) that populate the model. In this framework contracts are used to integrate the top-down specification of organizational structures with the autonomy of participating agents. In this paper we introduce LCR, a very expressive logic for describing interaction in multi-agent systems. We also show how LCR behaves in contrary-to-duty situations common to deontic logic frameworks. LCR makes it possible to check whether agents in an agent society follow some desired interaction patterns and whether desired social states are preserved by agent activity. LCR is used as a formal basis for the framework for agents societies that we are developing.

1 Introduction

Agent-based computing has been an active research topic for many years. Agent concepts gained relevance in industry as adequate means to describe and build large and complex systems. Due to their autonomous, pro-active and social behavior, agents can better adapt to changes in their environment and solve problems they encounter during operation with limited intervention from the user. This has a large advantage over traditional systems for which the environment of the system had to be completely predictable or otherwise the system would not function correctly. However, if one creates a system with a number of autonomous agents it becomes unpredictable what the outcome of their interactions will be. In settings where the multi-agent system is used to implement a system with specific goals, this so-called emerging behavior can be perceived as a problem, because one does not want this emergent behavior to diverge from the overall goal of the system. Furthermore, it is unrealistic to think that any directed behavior will happen from the fact that agents happen to share some environment. Like humans, software agents will not work together just because they happen to be together, but require some external incentives for collaboration. We call such systems with specific goals, organizational systems.

We have developed an agent-oriented model for organizational systems, the **Agent Society Model** that emerges from the idea that in an organizational system, as in any organized society, interactions between members occur not just by accident but aim at

achieving some desired global goals. That is, there are goals external to each individual agents that must be reached by the interaction of those agents. The desired global behavior of an organization is therefore external to the participating agents. Although agents will contribute to achievement of society goals, this happens only when such goals fit with the agents' own goals, or because of the agents' own motivation. Conceptually global organizational goals and rules cannot be attributed or modeled as part of the agents. Furthermore, we start from the fact that social structure is determined by organizational design and not dependent on the agents themselves.

The Agent Society Model distinguishes between the mechanisms though which the structure and global behavior of the model is described and coordinated, and the aims and behavior of the service-providers (agents) that populate the model [6, 16].

In order to represent interactions between agents in an open context, a framework is needed that is able to specify sequences of interaction scenes

- independently from the agent's internal design (**internal autonomy requirement**)
- without fixing the interaction structures completely in advance (**collaboration autonomy requirement**).

The framework consists of three interrelated models. The structure and coordination norms as intended by the organizational stakeholders are described in the **Organizational Model** (OM). Components of this model are roles, constraints, interaction rules, and communicative and ontology framework. Based on their own goals, individual agents join the society as enactors of roles. Possible agent populations of the organizational model are specified in the **Social Model** (SM) in terms of contracts that make explicit the commitments regulating the enactment of roles by individual agents. Finally, given an agent population, the **Interaction Model** (IM) describes possible interaction between agents. Depending on the aims and characteristics of the application, the OM will allow for more or less freedom for its agent population to decide and negotiate on how to interact with each other. In order to limit the unpredictability of the system that may arise due to the autonomous behavior of agents, agreements concerning role enacting and interaction are fixed in contracts. **Contracts** allow to integrate the top-down specification of organizational structures with the autonomy of participating agents. The use of contracts to describe activity of the system allows in one hand for flexibility in the balance between organizational aims and agent desires and on the other hand for verification of the outcome of the system.

In this paper we present a logical formalism for describing interaction in a agent society. The formalism enables the specification of social norms and interaction contracts. The logical formalism combines elements from deontic and branching time logic. In the remainder of this paper we will introduce the main features of this logic. The paper is organized as follows. In section 2 we give some rationale for the formalization of agent societies. Section 3 introduces LCR, the logic for contract representation that we have developed. In section 4 we show how several contrary-to-duty situations are handled in LCR. Section 5 described how LCR can be used to represent interaction contracts between agents. In section 6 we discuss related work on the formalization of organizational behavior on multi-agents systems. Finally, in section 7 we present our conclusions and indicate directions for further research.

2 Formal Specification of Agent societies

Norms have been identified in social sciences as crucial tools for important (agent) societies issues such as coordination, cooperation, trust and reputation. A formalism for agent societies must be able to uniformly describe and reason about social structure (landmarks and roles) and interaction (social and interaction contracts). Such formalism facilitates the analysis of societies and verification through logical reasoning, that is, verification of society design gets down to prove inconsistencies in the logical description.

In systems where agents are assumed to be autonomous and intelligent, agents can, involuntarily or by deliberate choice, violate social norms and regulations and therefore one must be able to deal and reason about violations. The use of deontic logic as a formalism for multi-agent systems has been advocated by several researchers (cf. [15]). Deontic logic provides mechanisms to reason about violability of norms, that is, about how to proceed when norms are violated. In practice, logical formalisms for agents have been used to (1) specify agents in an abstract manner and to (2) verify and reason about agent behavior, independently of the implementation language used to represent the agent.

A more advanced form of agenthood, normative agents (that is, agents that can reason about norms and obligations) can bridge the gap between individual autonomous agents and the agent society, in the sense that the cognitive concept of obligation is the building block of complex social notions like coordination, cooperation, trust and reputation.

Furthermore, verification of the behavior of an open society, where the design of participating agents cannot be controllable, must be based on the externally observable effects of agent actions. That is, from the society perspective, different actions that bring about the same state of affairs in the world cannot be distinguished. From the above considerations, it follows that a logical formalism for the Agent Society Model must be able to represent:

- Deontic relations (obligations, prohibitions, permissions)
- Externally observable results of agent actions (changes in state caused through influence of agents)
- Temporal relationships (effect of actions and agreements is not instantaneous and not deterministic, several futures are possible at each moment depending on agent decisions and environment changes)
- Violations and reasoning about effects and recovery from violated states

3 Logic for Contract Representation

The Logic for Contract Representation (LCR) that we propose is based on a branching-time logic. This means that formulae are interpreted over tree-type branching structures that represent all conceivable ways the system can evolve. Nodes represent *states* and arcs correspond to the occurrence of *events*. A *path* represents a course of events and links states in the time structure according to the choices and possibilities available to agents at each moment. Our proposal extends the formalism based on Temporal and Deontic Logic, BTLcont, proposed by Dignum and Kuiper [4]. BTLcont is in itself an extension to the well known branching-time temporal logic (CTL*) proposed by Emerson and Halpern [8, 10]. While Emerson and Halpern

provide a sound and complete axiomatization for CTL*, we do not address the issue of completeness in this paper. Our main aim is to present an expressive semantics for contracts, that represent interaction between agents in an abstract way, that is, independent from the internal architecture of the agents.

We further extend branching time logic with a *stit* operator, E_a ('agent a sees to it that') based on Pörn [14]. This allows us to refer to the externally 'observable' consequences of an action instead of the action itself. Remember that agent internals are not visible from the organizational perspective, and therefore it is not possible to refer to specific actions of an agent. In our use of E_a we draw from the logic proposed by Wooldridge for the combination of a *stit* operator with a temporal logic [17].

Moreover, clauses in a contract (deontic expressions in LCR) indicate that something must happen (ideally something happens) but in fact it may never happen at all! A logic for contract representation must therefore be able to reason about states in which an obligation has been violated. Obligations have to do with the preference of individuals (or societies) to be in a certain state. $O_a\varphi$ (the obligation for agent a to see to it that φ holds) indicates that, in the society, it is preferable for a to be in a state where φ holds rather than in any other state. This does not mean that agent a cannot be in other states either by choice or necessity. A violation, viol(a,φ,δ), is interpreted as 'agent a is in a violation situation concerning the obligation to do φ before deadline δ'[1]. The basic idea is that worlds in which a violation proposition holds are less preferred by the agent concerned. Sanctions are defined in order to make it possible for violations to be redeemed.

3.1 Syntax of LCR

LCR is an extension of CTL*, which in turn is an extension of classical propositional logic[2]. Well-formed formulae of LCR are built of a set Φ of atomic propositions that may be combined using the classical proposition connectives $\vee$ ('or') and $\neg$ ('not'). Other propositional connectives such as $\wedge$ ('and'), $\rightarrow$ (logical implication) and $\leftrightarrow$ (logical equivalence) can be introduced as abbreviations. The language also contains the constants *true*, *false*, the CTL* operators A (always in the future), S (since), X (in the next state), Y (yesterday, or in the previous state), U (until), $\leq$ (before) and the *stit* operator E. Furthermore, we introduce a predicate viol(a, φ, δ) that holds in states where $O_a(\varphi \leq \delta)$ has been violated by agent a. The E operator is labeled with agents and/or group identifiers. Elements `a, b, …,` of a set `Ags` of agent identifiers are used as labels for E. For example E_a is read as 'agent *a* sees to it that'.

Definition 1 Syntax of LCR

The set of well-formed formulae of LCR is introduced inductively, given a set Φ of atomic propositions (including *true*, *false*). As in CTL*, LCR distinguishes between state formulas (evaluated in a state) and path formulas (evaluated in a path).

1. Every member of Φ is a state formula
2. If φ, φ_1 and φ_2 are state formulas, then so are $\neg\varphi$, $\varphi_1 \vee \varphi_2$, $Y\varphi$ and φ_1 S φ_2

[1] When no deadline is specified for an obligation, the violation is simplified to viol(a, φ).

[2] In finite domains, the existential quantifier can be introduced as a finite disjunction and the universal quantifier as finite conjunction.

3. If φ is a state formula, then so is $E_a\varphi$, for all $a \in Ags$
4. If φ_1 and φ_2 are state formulas, then so is viol(a, φ_1, φ_2), for all $a \in Ags$
5. Each state formula is also a path formula
6. If ψ is a path formula, then Aψ is a state formula
7. If ψ, ψ_1, ψ_2 are path formulas, then so are $\neg\psi$, $\psi_1 \vee \psi_2$, $\psi_1 U \psi_2$, $\psi_1 \leq \psi_2$ and Xψ

3.2 Semantics of LCR

Usually different events are possible at any moment. That is, at each moment different futures are possible depending on the events in the world. We therefore have defined the semantics for LCR using branching time structures.

Definition 2 Branching Time Structure

A branching time structure is a tuple (W, R) where:

- W is a set of worlds (**states**) and
- $R \subseteq W \times W$ is the successor relation on states, such that the reflexive, transitive closure of R, R*, is a total tree relation.

R* represents all possible courses of system history. A **path** (or trace) through R is a sequence $(s_i \mid i \in \mathbb{N})$ such that $\forall i \in \mathbb{N}$ we have $(s_i, s_{i+1}) \in R$. If t is a path then state t(i) is the i-th element of p. We assume that there is a state s_0, which is the root of (W, R). Furthermore, we represent the tail of the path starting with state t(i) by t[i].

Definition 3 Semantic model

A semantic model M for LCR is a structure M = (W, R, π) where (W, R) is a branching time structure and π is a valuation function, which associates each $s \in W$ with the set of atomic propositions from Φ that are true in that world.

A path is a full and infinite sequence of states. Paths do not have to start from the root, but once started, there is always a following state in the path. By acting, agents can influence the next state in a path. The actions of agents are some of the possible events in the graph. In order to be able to represent the influence of an agent on changes in the world, we introduce the notion of controllable and uncontrollable expressions.

3.2.1 Controllable and non controllable propositions

Intuitively it only makes sense to specify $E_a\varphi$ for a formula φ if agent a can indeed 'see to it' that φ holds, that is if the agent can control or influence the truth value of φ. For instance, it does not make sense to express E_a*rains* because the fact whether it rains or not is not something that an agent can control. Inspired by the work of Boutelier [2] and Cholvy and Garion [4], we partition for each agent a the set of atomic propositions Φ in any world w of M in two classes: C_a and $\bar{C}_a$ in which C_a is the set of atomic propositions that agent a can control and $\bar{C}_a$ the set of atomic propositions that a cannot control.

Definition 4 Valuation function

1) Let π be the valuation function of a semantic model M = (W, R, π), which associates each s $\in$ W with the set of atomic propositions from Φ that are true in that world. For a set P of atomic propositions, **π(P)** indicates the restriction of π to the propositions in P (that is, the subset of true propositions of P). For every agent *a*, π can thus be written as $<\pi(C_a), \pi(\bar{C}_a)>$, the composition of the restriction of π to the controllable atomic propositions of *a* and the non-controllable atomic propositions of *a*.
2) For a set P of atomic propositions, **Π(P)** is the set of all valuations of atoms of P.
3) We need furthermore to define the concatenation of two valuation functions. Given two valuation functions u and v such that dom(u) $\cap$ dom(v) = $\varnothing$, **u.v(p)** is defined as:

$$u.v(p) = \begin{cases} u(p), p \in dom(u) \\ v(p), p \in dom(v) \end{cases}$$

Definition 5 Controllable and uncontrollable propositions

Given classes C_a and $\bar{C}_a$ defined as above,

1) a proposition φ is *a-controllable* in a state s $\in$ W, where M = (W, R, π) is a semantic model, iff $\forall u \in \Pi(\bar{C}_a)$, $\exists v_1, v_2 \in \Pi(C_a)$ and $\exists s_1 \in W$, $\pi(s_1) = u.v_1$, $\exists s_2 \in W$, $\pi(s_2) = u.v_2$, such that $(M, s_1) \models \varphi$ and $(M, s_2) \not\models \varphi$.
2) An expression is *a-uncontrollable* iff it is not a-controllable.

For example consider model M = (W, R, π) and the propositions p and q, such that $p \in C_a$ and $q \in \bar{C}_a$. In this case, proposition $p \wedge q$ is not a-controllable, because q is not a-controllable, and if q is false, agent a cannot make $p \wedge q$ to be true. Tautologies are never a-controllable, that is, if something is always true, no agent can claim to see to it that it will become true.

Theorem

$\forall\varphi$, if $\models \varphi$ then φ is a-uncontrollable for agent a. That is, agent a cannot control tautologies.

Proof. Suppose φ such that $\exists\varphi$, $\models\varphi$ and φ is a-controllable. $\models\varphi$ implies $\forall$s, (M,s)$\models \varphi$. However from the definition of a-controllable there must be a s $\in$ W such that (M, s) $\models \neg\varphi$. Therefore φ cannot be a tautology.

3.2.2 Path and State Semantics

As in CTL*, we define the semantics for state and path formulae separately. A path formula is a formula that is interpreted with respect to a path through a branching time structure. Paths correspond to histories of the system. In contrast, a state formula is interpreted with respect to a system state. The semantics of path formulae are given via the path formula satisfaction relation represented by '|=' that relates tuples of the

form (M, p), where M is a LCR-model, M = (W, R, π), and t a path (or trace) in M, to path formulae of LCR. This relation is defined by the following rules:

(P1) $(M, t) \models \varphi$ iff $(M, t(0)) \models \varphi$, where φ is a state formula
(P2) $(M, t) \models \neg\psi$ iff not $(M, t) \models \psi$
(P3) $(M, t) \models \psi_1 \vee \psi_2$ iff $(M,t) \models \psi_1$ or $(M, t) \models \psi_2$
(P4) $(M, t) \models X\psi$ iff $\forall$ t': $(t, t') \in R$, $(M, t') \models \psi$
(P5) $(M, t) \models \psi_1 U \psi_2$ iff $\exists i \in \mathbb{N}$ such that $(M, t[i]) \models \psi_2$ and $\forall k \leq i$, $(M, t[k]) \models \psi_1$
(P6) $(M, t) \models \psi_1 \leq \psi_2$ iff $\forall i \in \mathbb{N}$ such that $(M, t[i]) \models \psi_2$ and $\exists j \leq i$:, $(M, t[j]) \models \psi_1$

The semantics of state formulae are given via the state formula satisfaction relation, also represented by '|=' that relates tuples of the form (M, s), where M is a LCR-model, M = (W, R, π), and s a world in W, to state formulae of LCR. This relation is defined by the following rules:

(S1) $(M, s) \models p$ iff $p \in \pi(s)$, where $p \in \Phi$
(S2) $(M, s) \models \neg\varphi$ iff not $(M, s) \models \varphi$
(S3) $(M, s) \models \varphi_1 \vee \varphi_2$ iff $(M, s) \models \varphi_1$ or $(M, s) \models \varphi_2$
(S4) $(M, s) \models A\psi$ iff $\forall t \in$ *paths*(W, R), if $t(0) = s$ then $(M, t) \models \psi$
(S5) $(M, t(i)) \models Y\varphi$ iff $(M, t(i-1)) \models \varphi$
(S6) $(M, t(i)) \models \varphi_1 S \varphi_2$ iff $\exists k \leq i$ such that $(M, t(k)) \models \varphi_2$ and $\forall j, k < j \leq i$, $(M, t(j)) \models \varphi_1$
(S7) $(M, s) \models E_a\varphi$ iff 1) φ is *a-controllable*: $\forall s' \in W$, if $(s, s') \in R$, $(M, s') \models \varphi$
2) φ is *a-uncontrollable*: *false*

The semantics of LCR are standard branching time semantics with the exception of $E_a\varphi$. $E_a\varphi$ is intended to represent the fact that agent a sees to it that φ is satisfied. The semantic rule for $E_a\varphi$ can be described informally as: agent a acts in world w in such a way that the truth of the a-controllable expression φ is guaranteed. The *stit* operator E_a ignores the means by which agent a will bring about a state of affairs. We furthermore introduce the operator $D_a\varphi$ that represents the fact that a specific state of affairs has indeed been brought about by an agent in the previous world. $D_a\varphi$, meaning 'φ has been done by a' is defined as:

$$(M, t(i)) \models D_a\varphi \text{ iff } \quad (M, t(i-1)) \models \neg\varphi \wedge E_a\varphi$$

The following property holds for D_a:

$$\models D_a\varphi \rightarrow \varphi \qquad (1)$$

3.3 Representing deontic modalities in LCR

In a logic for the representation of contracts it must be possible to specify a time limit for realizing a certain state of affairs. Contracts express commitments that agents make to each other, that is an obligation for an agent to bring about a certain state of affairs (that is of interest to another agent). A deadline for the fulfillment of such obligations is usually indicated by the contract. A possible way to express deadlines is to indicate that an event should take place before a certain condition becomes true.

Moreover, clauses in a contract indicate that something must happen (it is desirable that something happens) but in fact if may never happen at all! A logic for contract representation must therefore be able to reason about situations (worlds in the semantics above) in which an obligation has been violated. Obligations have to do with the preference of individuals (or societies) to be in a certain state. $O_a\varphi$ indicates that, in the holding society, it is preferable for a to be in a state where φ holds than in any other state. This does not mean that agent a cannot be in other states either by choice or necessity. Worlds where a violation proposition holds are less preferred by the agent concerned.

We introduce obligation as a derived operator in LCR. Obligations in LCR express the fact that agent a is expected to bring about a certain state of a affairs (result) r before a certain condition (deadline) d has become valid.

Definition 6 Obligation with deadline:

The obligation of agent a to see to it that result ρ is achieve before an event δ happens, is defined in LCR as:

$$O_a(\rho \leq \delta) =_{def} A(\neg\delta \, U \, ((E_a\rho \wedge X(A\Box\neg viol(a,\rho,\delta))) \vee X(\delta \wedge viol(a,\rho,\delta))))$$

Where $X(A\Box \neg viol(a,\rho,\delta))$ indicates that this violation will not occur anymore in the future[3]. The obligation without deadline is a special case of the definition above:

$$O_a\rho =_{def} O_a(\rho \leq \mathit{true})$$

It is interesting to note that in LCR the proposition $O_a(\rho \leq \delta) \wedge O_a(\neg\rho \leq \delta)$ is consistent. As actually is $O_a\rho \wedge O_a\neg\rho$. One or both of the obligations can be true due to the fact that the violation for that obligation is true.

The definition of obligation expresses the fact that in all worlds reachable from a world where $O_a(\rho \leq \delta)$ holds either the agent has seen to it that result ρ has been achieved or a violation of the obligation holds in those worlds. Intuitively, the idea is that an obligation will 'disappear' once the result is achieved within the deadline. However, this is not the case. Fulfilling an obligation does not mean that the obligation disappears but, once the result is achieved within the deadline, the obligation can never result in a violation anymore. Formally this is represented as:

$$\models O_a(\rho \leq \delta) \rightarrow \neg viol(a, \rho,\delta) \; S \; D_a(\rho \leq \delta) \qquad \textbf{(2)}$$

Conditional obligations are obligations that only become active if the precondition becomes valid. Unlike regular obligations, that only hold once, a conditional obligation will come in force every time the condition holds.

Definition 7 Conditional Obligation with deadline

The obligation of agent a to see to it that result ρ is achieved before an event ρ happens given that precondition π holds, is defined in LCR as:

$$O_a(\rho \leq \delta \mid \pi) =_{def} A((\pi \rightarrow O_a(\rho \leq \delta)) \, U(D_a\rho \vee \delta))$$

[3] $\Box\varphi$ is defined as: $\Box\varphi =_{def} \neg(\mathit{true} \, U \, \neg\varphi)$.

In this definition, the expression $U(D_a\rho \vee \delta)$ is necessary in order for the conditional obligation to be removed once it has been realized (or it cannot be done anymore because the deadline has passed). Otherwise, whenever π becomes true the obligation will arise. Because π can still be true after the obligation is fulfilled, the obligation will arise again and again.

Note that the special case of a conditional obligation, $O_a(\rho \leq \delta \mid \mathit{true})$ is not the same as the regular obligation $O_a(\rho \leq \delta)$, but expresses an obligation that always holds. Another property of conditional obligations, is that they became 'normal' obligations whenever the precondition holds. Formally, this is expressed as:

$$\models O_a(\rho \leq \delta \mid \pi) \wedge \pi \wedge \neg\rho \wedge \neg Y\delta \rightarrow O_a(\rho \leq \delta) \qquad (3)$$

Intuitively, one expects that once a deadline has passed, its violation will always hold, or at least until a sanction is performed. However this is not yet the case, if we consider the definitions above. We need thus to introduce the following axiom:

Axiom $\models viol(a, \rho, \delta) \rightarrow A(\, viol(a, \rho, \delta)\ U\ D_a\sigma)$

where σ is the sanction that will remove viol(a, ρ, δ), which can also be represented by sanction(σ, a, ρ, δ).

Sanctions themselves can be seen as obligations conditional on the occurrence of a violation. This leads to the following observation concerning the sanctions. If sanction(σ, a, ρ, δ), that is, σ is the sanction that will remove viol(a, ρ, δ), then:

$$O_a(\sigma \leq \delta' \mid viol(a, \rho, \delta)) \wedge viol(a, \rho, \delta) \rightarrow \qquad (4)$$
$$((\neg viol(a, \rho, \delta) \wedge \neg viol(a, \sigma, \delta')) \vee ((viol(a, \rho, \delta) \wedge viol(a, \sigma, \delta'))S\delta'$$

So, either all the related violations disappear through performing the sanction or additional violations arise when it is not performed.

Obligations that hold in cases of violation of another obligation, such as sanctions, are known as contrary-to-duty obligations. Contrary-to-duty situations lead to some well-known paradoxes in standard deontic logic (SDL). In the next section we discuss some of these and describe how our formalism behaves in contrary-to-duty situations.

4 Contrary-to-duty Imperatives

A contrary-to-duty obligation is an obligation that is only in force in a sub-ideal situation. This is often necessary to represent some aspects of legal systems. Unfortunately, contrary-to-duty reasoning leads to notorious paradoxes of deontic logic. Paradoxes of deontic logic are logical expressions (in some logical language) that are valid in (many) well-known logical systems for deontic reasoning, but which are counterintuitive in a common sense reading [11]. The problem with most contrary-to-duty situations is that obligations referring to the most ideal situation conflict with obligations referring to less ideal cases. In contrast to many deontic logics, LCR explicitly represents the notion of violation. Intuitively, violation changes the context of (normative) reasoning. Violation contexts distinguish between ideal and sub-ideal contexts, varying in degree of 'ideality'. Therefore, the representation of contrary-to-duty imperatives in LCR is in most cases straightforward.

It is not our intention to show here how LCR behaves for all the many contrary-to-duty situations that have been described for deontic logic, but we will take three

versions of the Chisholm paradox [3], the forward, the parallel and the backward versions, as representative. Moreover, because our research is applied in the area of Knowledge Management and the support of Communities of Practice, we have taken examples from this areas for the informal description of the paradoxes.

Note that, from the definitions of obligation and conditional obligation in LCR, it can be proven that $O_a(\varphi) \wedge O_a(\psi|\varphi)$ does not imply $O_a(\psi)$. This is essential for the faithful representation of contrary-to-duty situations.

4.1 The forward version of the Chisholm paradox

In this contrary-to-duty situation extra activities are obliged after a certain obligation holds, which do not hold otherwise. In our example, the rules of a knowledge sharing community are as follows: Meeting chairs must publish notes of the meeting (F1). When meeting notes are published, this must be announced to group members (after publishing) (F2). If not published, then it must not be announced (F3).
In SDL, a paradox follows from the case that notes are not published (F4). The formal specification of the above rules, in the generic case is:

(F1) $O_a(\varphi)$ (a is obliged to publish meeting notes)
(F2) $O_a(\psi \mid \varphi)$ (given that notes are published, a is obliged to announce it)
(F3) $O_a(\neg\psi \mid \neg\varphi)$ (if not publish, a is obliged not to announce publication)
(F4) $\neg\varphi$

From the way obligations are defined in LCR, F1 expresses that φ still has to be made true by a, that is $Ea(\varphi)$, and in F2 the obligation $O_a\psi$ only holds in states where φ already holds. This implies a time difference between when φ should be true and ψ should be true. This is why this represents the forward version of the Chisholm paradox. Because, in our formalism, violation of norms is explicitly represented, this paradox does not result in states where a contradiction holds. Originating states can, however, be associated with a preference. The following table portrays the different states originating in this situation, where state S0 represents a state where the obligations hold and no action has been done and in each state only the relevant propositions are specified.

Table 1. Forward version of Chisholm paradox in LCR

S_0		**S_1**		**Possible next states**	
$O_a(\varphi)$ $O_a(\psi \mid \varphi)$ $O_a(\neg\psi \mid \neg\varphi)$	*(F1)* *(F2)* *(F3)*	$\neg\varphi$ viol(a, φ) $O_a(\neg\psi)$	*(F4)* *(from F1)* *(from F3)*	$\neg\varphi$ ψ viol(a, φ) viol(a,ψ)	*(F4)* *(from F1)* *(from F3)*
				$\neg\varphi$ $\neg\psi$ viol(a, φ)	*(F4)* *(from F1)*

4.2 The parallel version of the Chisholm paradox

This version of the Chisholm paradox is better known as the Forrester, or 'gentle murder' paradox. An example of this contrary-to-duty situation, that results in a paradox in SDL, is when the following. agreements hold in a community: Members of the community are not allowed to publish internal department reports (P1). But, if a

member does publish an internal report, then it must be published in the discussion area (P2). Because publishing in the discussion area is a special case of publishing, this means that both activities are simultaneous (P3). The paradox arises when a report is published (P4). The generic formal specification of this situation is:

(P1)	$O_a(\neg\varphi)$	(a is obliged not to publish internal reports)
(P2)	$O_a(\varphi \rightarrow \psi)$	(if published, a must publish it in the discussion area)
(P3)	$\psi \rightarrow \varphi$	(publishing follows from publishing in discussion area)
(P4)	φ	(a report is published)

Because, in our formalism, violation of norms is explicitly represented, this situation does not result in states where a contradiction holds. From P1 to P3, it is not possible in LCR to derive $O_a(\psi)$, which is usually the cause of the 'gentle murder' paradox in SDL. The following table portrays the different states originating in this situation, where state S0 represents a state where the obligations hold and no action has been done and in each state only the relevant propositions are specified. Because ψ can only be done at the same moment as φ, in states where φ holds either ψ or $\neg\psi$ must hold (ψ cannot happen at a latter state).

Table 2. Parallel version of Chisholm paradox in LCR

S_0		Possible next states	
		φ	*(P4)*
		ψ	
$O_a(\neg\varphi)$	*(P1)*	viol(a, $\neg\varphi$)	*(from P1)*
$O_a(\varphi \rightarrow \psi)$	*(P2)*	φ	*(P4)*
$\psi \rightarrow \varphi$	*(P3)*	$\neg\psi$	
		viol(a, $\neg\varphi$)	*(from P1)*
		viol(a, $\varphi \rightarrow \psi$)	*(from P2)*

4.3 The backward version of the Chisholm paradox

In this contrary-to-duty situation extra activities are obliged before a certain obligation holds, which are not obliged. In our example, this can be described by the situation in which the following community rules hold: Members must attend group meetings (B1). If one attends a meeting, then one must tell that one is coming (before) (B2). If one does not attend a meeting, then one must not tell that one is coming (B3). In SDL a paradox will occur when one does not attend a meeting (B4). The generic formal specification of this situation is:

(B1)	$O_a(\varphi)$	(a is obliged to attend group meetings)
(B2)	$O_a(\psi \leq \varphi)$	(If a attends, a must tell a is coming, before the meeting)
(B3)	$O_a(\neg\psi \leq \neg\varphi)$	(If a doesn't attend, a must not tell a is coming)
(B4)	$\neg\varphi$	

From this specification, one can see that LCR is utmost suitable to represent backward contrary-to-duty obligations due to the fact that in LCR deadlines are used explicitly. Because, in LCR, violation of norms is explicitly represented and there is a clear notion of time, the above situation does not result in states where a contradiction holds. The following table portrays the different states originating in this situation, where state S0 represents a state where the obligations hold and no action has been done and in each state only the relevant propositions are specified. Because ψ must be done before φ, in states where φ holds (B4) either ψ or $\neg\psi$ must already hold.

Table 3. Backward version of Chisholm paradox in LCR

S_0		Possible next states	
		$\neg\varphi$	*(B4)*
		$\neg\psi$	
$O_a(\varphi)$	*(B1)*	viol(a, φ)	*(from B1)*
$O_a(\psi \leq \varphi)$	*(B2)*	$\neg\varphi$	*(B4)*
$O_a(\neg\psi \leq \neg\varphi)$	*(B3)*	ψ	
		viol(a, φ)	*(from B1)*
		viol(a, $\neg\psi$, $\neg\varphi$)	*(from B3)*

5 Using Contracts to Model Interaction

The interaction structure as specified in the OM, gives a partial ordering of interaction scenes and the consequences of transitions from one to the other scene to the roles (creation, destruction, etc). That is, the interaction structure indicates the possibilities of an agent at each stage and the consequences of its choices. Explicit representation of commitments is necessary in order to know how to proceed if the norm is violated but first of all to inform the agents about the behavior they can expect from the other agents. In this way, coordination can become possible. A contract is a statement of intent that regulates behavior among organizations and individuals. A formal language for contract specification, such as LCR, allows the evaluation and verification of agent interaction.

Definition 8 Contract.

We define a **contract**, **C**, as a tuple **C** = (A, CC, S, T) where A is a set of agents, CC is a set of contract clauses (expressed as LCR formulae), S is the possible **stages** of the contract, and T gives the **transition rules** between states.

Transition rules are events in the domain that alter the status of the contract. Alternatively, a **stage graph** can be used to describe allowed states of the contract and the transition rules between states. The state graph represents the possible evolution(s) of the contract and the consequences of the changes in state to the different parties. Contract clauses are deontic expressions of LCR, that is, obligations, permissions or prohibitions, and as such may indicate deadlines and/or conditions.

Example

The following example is intended to illustrate the use of LCR to represent the interactions between two agent in a gent society. This contract expresses an exchange commitment agreed between agents S (a seller) and B (a buyer): S has agreed to sell B a bicycle for €500. S has 2 days to give the bicycle to B after which B must pay S within 1 day. If S does not provide the bicycle on time, then the exchange will not go through. If B does not pay on time then an extra €10 is due within 2 days. Formally, the clauses of this contract can be specified in LCR as[4]:

1. O_S(get-goods(B, bicycle) ≤ 2 days)

[4] Note that, for example, '2 days' abbreviates the expression that denotes a time 2 days from now (the evaluation moment), and is true only at that point in time.

2. O_B(get-money(S, €500) ≤ 1 day | D_S(get-goods(B, bicycle) ≤ 2 days))
3. O_B(cancel-deal(S,B, bicycle, €500) ≤ 1 day | viol(S,get-goods(B,bicycle),2 days))
4. O_B(get-money(S, €510) ≤ 2 days | viol(B, get-money(S, €500),1 day))

The contract transition graph depicted in Figure 1, represents the possible evolution of the contract and the consequences of the changes to the different parties. Transitions between contract stages are expressions representing events in the agent society. In the figure, arrows labeled *r* represent the achievement of the result and arrows labeled *d* represent that the deadline has passed without result. In each box only the relevant propositions are displayed. A bold box is the initial stage and double lined boxes are final stages. Stages S6 and S8 are not specified in the contract. Since most contracts are not exhaustive, such stages will probably appear in every contract graph. Consequences of reaching such stage can be determined by society norms (for example, guilty agent is expelled from society).

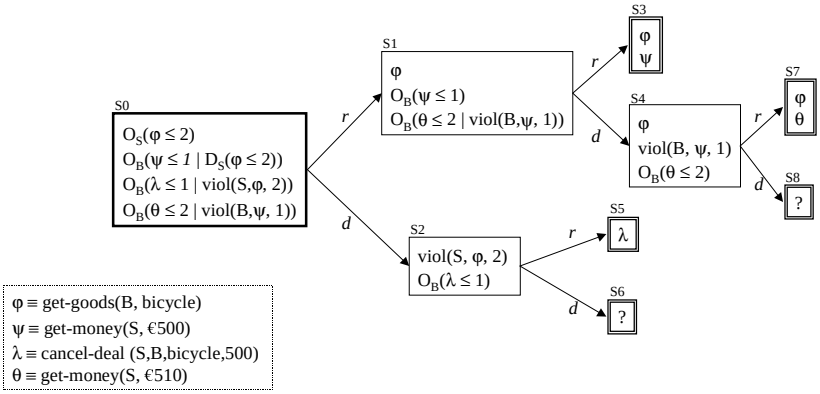

Figure 1: Example of a contract

6 Related Work

A main question to which this paper contributes is how formalize the behavior of multi-agent systems, and how to relate this behavior to the global objectives of the system. Commitments play an important part in agent interactions. In situations where several agents cooperate within an organizational framework, designed to realize global society objectives, commitments are a means to specify expectations on the behavior of other participants. Several approaches to the formalization of interaction have been presented that are based purely in terms of organization concepts (and thus not referring to specific agents).

Behavioral approaches to the design of multi-agent systems are gaining terrain in agent research and several research groups have presented models similar to our proposal. Recent developments recognize that the modeling of interaction in MAS cannot simply rely on the agent's own (communicative) capabilities. Furthermore, organizational engineering of MAS cannot assume that participating agents will act according to the needs and expectations of the system design. Concepts as organizational rules [18], norms and institutions [7] and social structures [13] all start from the idea that the effective engineering of MAS needs high-level, agent-

independent concepts and abstractions that explicitly define the organization in which agents live [19].

One of the first works in this area is that of Ferber and Gutknecht [9]. The organisation model structure they use includes high level concepts such as groups and roles within groups, and (intragroup and intergroup) role interaction. However expressive, AALAADIN does not offer primitives to describe interaction and coordination within and between groups and agents and the environment. This model was used as basis for a proposal for representation of social structures in AUML that does describe interaction between roles [13].

The model developed by the Alfebiite consortium is meant for the design of open agent societies and considers aspects of security and legal consequences of agent action in agent societies [1]. The model includes representation primitives for agents, constraints, communication language, roles, social states and agent owners. In our opinion, this model lacks primitives for the representation of groups and complex interaction and coordination in a society.

Esteva et al. [7] devise a formal specification language to design open agent systems as *electronic institutions* with focus on the normative aspects of societies. This proposal aims at the modeling of institutionalized electronic organizations (institutions). In this approach, roles are defined as patterns of behavior, normative rules are specified to limit or enlarge the space of agent actions and scenes are defined in order to represent the different contexts within an organization in which agents can interact. However, this framework takes a very low level approach to abstract interaction, by demanding that all interaction be expressed in terms of fully specified protocols.

A recent approach, based on deontic logic, is that of Pacheco and Carmo [12]. They propose a role based model for *organized collective agency*, based on the legal concept of *artificial person* and on a normative perspective on organizations. Their logic attempts to capture the concept of taking up a role. However, the logic does not include any temporal concepts, which makes it not suitable to represent real life organizations. Moreover, they lack a formal definition of roles (viewed as identifiers) and assume that roles are generated from the contracts between agents.

7 Conclusions

Contracts are used in an agent society to indicate agent conformance to some desired interaction patterns and to verify whether desired social states are preserved by agent activity. In this paper we have introduced LCR, a very expressive logic for describing interaction in multi-agent systems. This logic makes it possible to describe and verify contracts that specify interaction between agents. LCR is used as a formal basis for the framework for agents societies that we are developing. So far, we have concentrated on the logical representation of contracts. Future work needs to investigate how to reason formally about the interaction structure as a whole.

Acknowledgement. The authors wish to thank Andrew Jones for his valuable comments on a previous version of this paper presented at the 2nd FAABS workshop.

References

1. Artikis, A., Pitt, J.: A Formal Model of Open Agent Societies. Proc. Autonomous Agents 2001, (2001) 192-193
2. Boutelier, C.: Toward a Logic for Qualitative Decision Theory. In: Doyle, J., Sandewall, E., Torasso, P. (Eds.): Principles of Knowledge Representation and Reasoning, (1994).
3. Chisholm, R.:Contrary-to-Duty Imperatives and Deontic Logic.Analysis 24,(1963), 33-36.
4. Cholvy L., Garion C.: An Attempt to Adapt a Logic of Conditional Preferences for Reasoning with Contrary-To-Duties. In: Fundamenta Informaticae, 47 (2001).
5. Dignum,F., Kuiper, R.: Specifying Deadlines with Dense Time using Deontic and Temporal Logic. In: International Journal of Electronic Commerce, 3(2), (1999), 67 – 86.
6. Dignum, V., Meyer, J.-J., Weigand, H., Dignum, F.: An Organizational-oriented Model for Agent Societies. In: Proc. Int. Workshop on Regulated Agent-Based Social Systems: Theories and Applications (RASTA'02), at AAMAS, Bologna, Italy, July, (2002).
7. Esteva, M., Rodriguez, J., Sierra, C., Garcia, P., Arcos J.: On the formal specifications of electronic institutions, In Dignum F., Sierra C. (Eds.): Agent-mediated Electronic commerce (The European AgentLink Perspective), LNAI 1991, (2001), 126-147.
8. Emerson, E.: Temporal and Modal Logic. In: J. van Leeuwen (Ed.): Handbook of Theoretical Computer Science, Elsevier Science Publishers (1990).
9. Ferber, J., Gutknecht, O.: A meta-model for the analysis and design of organizations in multi-agent systems. Proc. of ICMAS'98, IEEE Press, 1998.
10. Halpern J., Moses, Y.: A guide to the completeness and complexity of modal logic of knowledge and belief. Artificial Intelligence 54(3), (1992), 319-379.
11. Meyer, J.-J., Wieringa, R., Dignum, F.: The Role of Deontic Logic in the Specification of Information Systems. In Chomicki J., Saake, G. (eds.): Logics for Databases and Information Systems, pages 71-115, Kluwer Academics Publishers, (1998).
12. Pacheco O., Carmo, J.: A Role Based Model for the Normative Specification of Organized Collective Agency and Agents Interaction. Journal of Autonomous Agents and Multi-Agent Systems, to be published (2003).
13. Parunak, H. V. D. and Odell, J.: Representing Social Structures in UML. In: Wooldridge, M., Weiss, G., Ciancarini P. (Eds.): Agent-Oriented Software Engineering II, LNCS 2222, Springer-Verlag, (2002) 1 - 16.
14. Pörn, I.: Some Basic Concepts of Action. In: Stenlund, S. (Ed.): Logical Theory and Semantic Analysis. Reidel, Dordrecht (1974).
15. Van der Torre, L.: Contextual Deontic Logic: Normative Agents, Violations and Independence. In: Annals of Mathematics and Artificial Intelligence, Special Issue on Computational Logic in Multi-Agent Systems, 37(1-2): 33-63, January 2003.
16. Weigand, H., Dignum, V., Meyer, J.-J.: Specification by refinement and agreement: designing agent interaction using landmarks and contracts. In: Proceedings ESAW'02, Madrid, Spain, (2002).
17. Wooldridge, M.: Time, Knowledge, and Choice . In: M. Wooldridge, J. P. Mueller, M. Tambe, (Eds.): Intelligent Agents II, Springer-Verlag, (1996).
18. Zambonelli, F.: Abstractions and Infrastructures for the Design and Development of Mobile Agent Organizations. In: Wooldridge, M., Weiss, G., Ciancarini P. (Eds.): Agent-Oriented Software Engineering II, LNCS 2222, Springer-Verlag, (2002), 245 – 262.
19. Zambonelli, F., Jennings, N., Wooldridge, M.: Organizational Abstractions for the Analysis and Design of Multi-agent Systems. In: Ciancarini, P., Wooldridge, M. (Eds.): Agent-Oriented Software Engineering, LNCS 1957, Springer, (2000), 235 – 251.