Comput Math Organiz Theor (2006) 12:205–225
DOI 10.1007/s10588-006-9544-8

Mapping deontic operators to abductive expectations

Marco Alberti · Marco Gavanelli · Evelina Lamma · Paola Mello · Giovanni Sartor · Paolo Torroni

Abstract Deontic concepts and operators have been widely used in several fields where representation of norms is needed, including legal reasoning and normative multi-agent systems.

The EU-funded SOCS project has provided a language to specify the agent interaction in open multi-agent systems. The language is equipped with a declarative semantics based on abductive logic programming, and an operational semantics consisting of a (sound and complete) abductive proof procedure. In the SOCS framework, the specification is used directly as a program for the verification procedure.

In this paper, we propose a mapping of the usual deontic operators (obligations, prohibition, permission) to language entities, called expectations, available in the SOCS social framework. Although expectations and deontic operators can be quite different from a philosophical viewpoint, we support our mapping by showing a similarity

M. Alberti · M. Gavanelli · E. Lamma
ENDIF, Università di Ferrara—Via Saragat, 1-44100 Ferrara, Italy
e-mail: malberti@ing.unife.it

M. Gavanelli
e-mail: mgavanelli@ing.unife.it

E. Lamma
e-mail: elamma@ing.unife.it

P. Mello · P. Torroni
DEIS, Università di Bologna—Viale Risorgimento, 2-40136 Bologna, Italy
e-mail: pmello@deis.unibo.it

P. Torroni
e-mail: ptorroni@deis.unibo.it

G. Sartor
CIRSFID, Università di Bologna—Via Galliera, 2-40100 Bologna, Italy
e-mail: sartor@cirfid.unibo.it

between the abductive semantics for expectations and the Kripke semantics that can be given to deontic operators.

The main purpose of this work is to make the computational machinery from the SOCS social framework available for the specification and verification of systems by means of deontic operators.

Keywords Normative systems · Deontic logic · Abduction · Semantics

1. Introduction

In the context of multiagent systems (MAS) the concepts of norms, commitments and social relations have been widely studied (Conte, Falcone and Sartor 1999). Furthermore, a lot of research has been devoted in proposing architectures for developing agents with social awareness (see, for instance, Castelfranchi et al., 1999), and several approaches to agent society modeling have applied a normative and institutional approach (e.g., Dignum et al., 2002a,b; Dignum, Meyer and Weigand, 2002c; Esteva, de la Cruz and Sierra, 2002; Noriega and Sierra, 2002).

Deontic logic (see Wright, 1951) has appeared a powerful reasoning tool for such approaches. In fact, it provides a simple formalism for explicitly and formally defining norms and dealing with their possible violations (see, for instance, van der Torre, 2003). It represents norms, obligations, prohibitions and permissions, and enables one to deal with predicates like "*p* ought to be done", "*p* is forbidden to be done", "*p* is permitted to be done".

In the context of the UE IST Programme, a whole research project, ALFEBIITE (1999) has been focused on the formalization of an open society of agents using Deontic Logic. In particular, the ALFEBIITE approach (presented, for instance, by Artikis, Pitt and Sergot, 2002) consists of a theoretical framework for providing executable specifications of particular kinds of multi-agent systems, called open computational societies, and presents a formal framework for specifying, animating and ultimately reasoning about and verifying the properties of systems where the behavior of the members and their interactions cannot be predicted in advance.

Another EU IST Project, SOCS (2002), has proposed a logic based approach to MAS from a different point of view, i.e., one especially oriented toward computational aspects: one of the main purposes of the social framework proposed by the project was to provide a computational framework to be directly used for automatic verification of properties, such as compliance to interaction protocols. The SOCS social model represents social rules in an abductive logic framework, where abducibles express expectations (positive and negative) on the behavior of members of the society; the semantics of the framework is based on abduction (Alberti et al., 2003). Operationally, the application of the abductive integrity constraints by an abductive proof procedure, called $\mathcal{S}$CIFF (Alberti et al., 2005b), adjusts the set of social expectations as the social infrastructure acquires new knowledge from the environment in terms of happened social events. $\mathcal{S}$CIFF has been proven sound and complete with respect to the declarative semantics, and has been implemented and integrated in a software component for the verification of open multi-agent systems (Alberti et al., 2006).

The subject of this work is a mapping of the commonly considered deontic operators (obligation, prohibition, permission) to the abductive expectations provided by the SOCS social framework: for example, an obligation can be mapped to a positive expectation (i.e., an event that is expected to happen). While this mapping can be hard to argument from a philosophical point of view, we support the mapping by showing a similarity between the Kripke semantics that has been given to deontic operators and the abductive semantics of expectations.

With this mapping, a sistem specified in the language of the SOCS framework can be understood with a deontic meaning. The purpose of this work is to make the computational machinery provided by the SOCS social framework available for the verification of systems (deontically) specified in such language.

In Section 2, we briefly recall the the SOCS social framework and deontic operators; in Section 3, we show the mapping of deontic operators to abductive expectations. In Section 4 we discuss some related work, and in Section 5 we conclude the paper.

2. Background

In this section, we briefly recall the aspects of the SOCS social framework and of deontic operators that are more relevant to the subject of the paper.

2.1. The SOCS social framework

The SOCS social framework (Alberti et al., 2005a) is aimed at the specification and verification of agent interaction in open agent societies.[1] Since in open societies no assumption can be made on the internal structure or attitude of the agents, the framework abstracts away from the individual agents (although the SOCS project also provides a model for individual agents, see for example Bracciali et al. (2005), and its extension to support normative agents recently proposed by Sadri et al. (2005)), and it rather constrains the *observable* agent behaviour.

For the specification, we proposed a language based on abductive logic programming (Kakas, Kowalski and Toni, 1993), equipped with an abductive declarative semantics. The language is not exclusively tailored on specification of agent behaviour (see, for example, Alberti et al., 2005b): however, its main application so far have been the specification of social semantics of agent communication (see, for example, Alberti et al., 2003) and the specification of agent interaction protocols (see, for example, Alberti et al., 2003c).

The verification of compliance (i.e., a check whether the agents are behaving accordingly to the specification) is brought about by means of the operational semantics of the language, consisting of the $\mathcal{S}$CIFF abductive proof procedure (Alberti et al., 2005b). The proof procedure has been proven sound and complete with respect to the declarative semantics. Thanks to the logic programming approach of the framework, the specification of the system is also used as a program for its verification. $\mathcal{S}$CIFF has been integrated in a software component for the verification of compliance to agent interaction protocols (Alberti et al., 2006).

[1]For a definition of openness, see Artikis, Pitt and Sergot (2002) and Hewitt (1991).

In this section, we briefly recall the abductive language used in the SOCS social framework, and its declarative semantics. For a detailed description, the reader can refer to Alberti et al. (2003).

2.1.1. Language

The SOCS language is aimed to:

1. describe the actual and desired agent behaviour;
2. specify the desired agent behaviour.

Description. In the following, we describe the intuitive meaning of the entities (events and expectations) used to describe the agent behaviour. Their formal semantics is given in Section 2.1.2.

The description of the actual agent behaviour is contained in a *history* **HAP**, i.e., a set of *events*. Events are represented as ground atoms of the form

$$\mathbf{H}(\mathit{Event}[,\mathit{Time}]),$$

where the *Event* argument is a ground term representing a description of the happened event, and the (optional) *Time* argument is an integer number representing the time at which the event happened.

For example, the following event

$$\mathbf{H}(\mathit{tell}(\mathit{alice}, \mathit{bob}, \mathit{query\_ref}(\mathit{phone\_number}), \mathit{dialog\_id}), \mathit{10}) \quad (1)$$

could represent the fact that *alice* asked *bob* his *phone_number* with a *query_ref* message, in the context identified by the constant *dialog_id*, at time 10.

Expectations are of the form

$$\mathbf{E}(\mathit{Event}[,\mathit{Time}]) \qquad \mathbf{EN}(\mathit{Event}[,\mathit{Time}])$$

for, respectively, positive and negative expectations. **E** is a positive expectation about an event (the event is expected to happen) and **EN** is a negative expectation (the event is expected not to happen). Explicit negation ($\neg$) can be applied to expectations. Differently from events, expectations can contain variables.

For example, the atom

$$\mathbf{E}(\mathit{tell}(\mathit{bob}, \mathit{alice}, \mathit{inform}(\mathit{phone\_number}, \mathit{Answer}), \mathit{dialog\_id}), \mathit{Ti}) \quad (2)$$

could represent an expectation for *bob* to *inform alice* that the value for the piece of information identified by *phone_number* is *Answer*, in the context identified by *dialog_id*, at time Ti.

Specification. A social specification $\mathcal{S}$ is composed of two parts:

- a social knowledge base KB_s;
- a set $\mathcal{IC}_s$ of *social integrity constraints*.

The Social Knowledge Base is a logic program, extended in that the body of the clauses can contain expectation literals. Intuitively, the social knowledge base can be used to express declarative knowledge about the agent society, ranging from simple information such as the value of time parameters, to complex organisational knowledge such as that regarding roles.

Specification 2.1 Social Knowledge Base for the *query_ref* social specification

$$qr\_deadline(10).$$

For example, Spec. 2.1 shows a simple example of a social knowledge base, which defines the *qr_deadline/1* predicate by means of one fact.

Social Integrity Constraints (also ICs, for short, in the following) are implications that, operationally, are used as forward rules by the $\mathcal{S}$CIFF proof procedure (Alberti et al., 2005b). Declaratively, their main use is to specify that is some set of events happens, then one of several other sets of events is expected to happen, or not to happen.

Specification 2.2 Integrity Constraints for the *query_ref* social specification.

$$
\begin{array}{rl}
 & \mathbf{H}(tell(A, B, query\_ref(Info), D), T) \ \wedge \\
 & qr\_deadline(TD) \\
\rightarrow & \mathbf{E}(tell(B, A, inform(Info, Answer), D), T1) \ \wedge \\
 & T1 \ < \ T + TD \\
\vee & \mathbf{E}(tell(B, A, refuse(Info), D), T1) \ \wedge \\
 & T1 \ < \ T + TD
\end{array}
$$

$$
\begin{array}{rl}
 & \mathbf{H}(tell(A, B, inform(Info, Answer), D), Ti) \\
\rightarrow & \mathbf{EN}(tell(A, B, refuse(Info), D), Tr)
\end{array}
$$

Spec. 2.1.1 shows the ICs for the *query_ref* social specification.

Intuitively, the first IC means that if agent A sends to agent B a *query_ref* message, then B is expected to reply with either an *inform* or a *refuse* message by TD time units later, where TD is defined in the Social Knowledge Base by the *qt_deadline* predicate (with the example in Spec. 2.1, the value of TD would be 10).

The second IC means that, if an agent sends an *inform* message, then it is expected not to send a *refuse* message at any time.

2.1.2. Declarative semantics

The semantics of a social specification in the SOCS social framework is of abductive type.

The following definition identifies a particular instance of a society, given by a social specification and a history of events.

Definition 2.1. Given a social specification $\mathcal{S} = \langle KB_S, \mathcal{IC}_S \rangle$ and a history **HAP**, $\mathcal{S}_{\mathbf{HAP}}$ represents the pair $\langle \mathcal{S}, \mathbf{HAP} \rangle$, called the **HAP**-*instance of* $\mathcal{S}$.

The following definition implements explicit negation (Apt and Bol, 1994) for expectation atoms.

Definition 2.2. A set **EXP** of expectations is $\neg$-*consistent* if and only if for each (ground) term p:

$$\{\mathbf{E}(p), \neg\mathbf{E}(p)\} \not\subseteq \mathbf{EXP} \qquad \text{and} \qquad \{\mathbf{EN}(p), \neg\mathbf{EN}(p)\} \not\subseteq \mathbf{EXP}. \tag{3}$$

The following definition prevents the same event from being both expected to happen and expected not to happen.

Definition 2.3. A set **EXP** of expectations is **E**-*consistent* if and only if for each (ground) term p:

$$\{\mathbf{E}(p), \mathbf{EN}(p)\} \not\subseteq \mathbf{EXP} \tag{4}$$

The following definition establishes a link between the actual and the expected agent behaviour, by requiring positive expectations to be matched by events, and negative expectations not to be matched by events.

Definition 2.4. Given a history **HAP**, a set **EXP** of expectations is **HAP**-*fulfilled* if and only if

$$Comp(\mathbf{HAP} \cup \mathbf{EXP}) \cup CET \cup \{\forall p\ \mathbf{E}(p) \rightarrow \mathbf{H}(p), \mathbf{EN}(p) \rightarrow \text{not } \mathbf{H}(p)\} \not\models false \tag{5}$$

where *Comp* represents the *completion* of a theory (Kunen, 1987), and *CET* is Clark's equational theory (Clark, 1978).

Otherwise, **EXP** is **HAP**-*violated*.

When **HAP** is apparent from the context, we will often omit mentioning it.

The following definition requires consistence of the set of expectations, with respect to an instance of the social specification.

Definition 2.5. Given a social specification $\mathcal{S} = \langle KB_s, \mathcal{IC}_s \rangle$, and an instance $\mathcal{S}_{\mathbf{HAP}}$ of $\mathcal{S}$, a set **EXP** of expectations is $\mathcal{S}_{\mathbf{HAP}}$-*consistent* if and only if

$$Comp(KB_s \cup \mathbf{HAP} \cup \mathbf{EXP}) \cup CET \models \mathcal{IC}_s \tag{6}$$

The following definition supports goal-directed social specifications: it requires the instance of the specification to entail a goal, while being consistent with respect to the previous definitions.

Definition 2.6. Given a social specification $\mathcal{S} = \langle KB_s, \mathcal{IC}_s \rangle$ and an instance $\mathcal{S}_{\mathbf{HAP}}$ of $\mathcal{S}$, a goal $\mathcal{G}$ is *achieved* in $\mathcal{S}_{\mathbf{HAP}}$ if there exists a $\neg$-consistent, **E**-consistent, $\mathcal{S}_{\mathbf{HAP}}$-consistent and **HAP**-fulfilled set **EXP** of expectations such that

$$Comp(KB_s \cup \mathbf{EXP}) \cup CET \models \mathcal{G} \tag{7}$$

In this case, we write $\mathcal{S}_{\mathbf{HAP}} \vDash_{\mathbf{EXP}} \mathcal{G}$ and we say that **HAP** is *compliant* to $\mathcal{S}$ with respect to $\mathcal{G}$.

In the remainder of this article, when we simply say that a history **HAP** is compliant to a social specification $\mathcal{S}$, we will mean that **HAP** is compliant to $\mathcal{S}$ with respect to the goal *true*. This is usually the case when the specification is used to express an interaction protocol, with no particular social goal. We say that a history **HAP** *violates* a specification $\mathcal{S}$ to mean that **HAP** is not compliant to $\mathcal{S}$.

The following definition identifies ill-defined social specifications, i.e., those for which there is no compliant history, which are obviously undesirable from an agent society designer viewpoint.

Definition 2.7. Given a goal $\mathcal{G}$, a social specification $\mathcal{S}$ is well-defined with respect to $\mathcal{G}$ iff there exists at least one history that is compliant to $\mathcal{S}$ w.r.t. $\mathcal{G}$, i.e., iff:

$$\exists \mathbf{HAP}\ \exists \mathbf{EXP} \mathcal{S}_{\mathbf{HAP}} \vDash_{\mathbf{EXP}} \mathcal{G} \tag{8}$$

When we simply say that a social specification $\mathcal{S}$ is well defined, we will mean that $\mathcal{S}$ is well defined with respect to the goal *true*.

Example 2.8. The *query_ref* social specification $\mathcal{S} = \langle KB_s, \mathcal{IC}_s \rangle$, where KB_s is defined in Spec. 2.1, and $\mathcal{IC}_s$ is defined in Spec. 2.2, is well defined. For instance, the history

$$\begin{gathered}\{\mathbf{H}(tell(alice, bob, query\_ref(phone\_number), dialog\_id), 10),\\ \mathbf{H}(tell(bob, alice, inform(phone\_number, 5551234), dialog\_id), 12)\}\end{gathered} \tag{9}$$

is compliant to $\mathcal{S}$.

One important observation to make at this point is that the semantics of our framework is based on an "all-or-nothing" concept of compliance: a history is either compliant to a specification, or it is not. Currently, we do not support *contrary-to-duty* obligations (Prakken and Sergot, 1996), exceptions, or any other kind of variable-degree compliance. We further discuss this issue, also considering related work, in Section 4.

2.2. Deontic operators

The birth of modern Deontic Logic can be traced back to the '50s (Wright, 1951). In the following, we only address the logical properties that are most useful in modeling legal reasoning, and norms, and refrain from addressing the logical background which provides a foundation for those properties.

Deontic Logic enables to address the issue of explicitly and formally defining norms and dealing with their possible violation. It represents norms, obligations, prohibitions and permissions, and enables one to deal with predicates like "*p* ought to be done", "*p* is forbidden to be done", "*p* is permitted to be done".

Being obligatory, being forbidden and being permitted are indeed the three fundamental *deontic statuses* of an action, upon which one can build more articulate normative conceptions. For details, refer to Sartor (2004), Chapter 15 in particular.

Obligations. To say that an action is *obligatory* is to say that the action is due, has to be held, must be performed, is mandatory or compulsory. Obligations are usually represented by formulas as:

$$\mathbf{Obl}\ A$$

where A is any (positive or negative) action description, and **Obl** is the deontic operator for obligation to be read as "it is obligatory that".

Elementary obligations can be distinguished between:

- *elementary positive obligations*, which concern positive elementary actions (e.g., "It is mandatory that John answers me");
- *elementary negative obligations*, which concern negative elementary actions (e.g., "It is mandatory that John does not smoke");

Prohibitions. The idea of obligation is paralleled with the idea of *prohibition*. Being forbidden or prohibited is the status of an action that should not be performed. In common language, and legal language as well, prohibitive propositions are expressed in various ways. For example, one may express the same idea by saying "It is forbidden that John smokes", "John must not smoke", "There is a prohibition that John smokes", and so on.

Prohibitions are usually represented by formulas as:

$$\mathbf{Forb}\ A$$

where A is any (positive or negative) action description, and **Forb** is the deontic operator for prohibition to be read as "it is forbidden that".

The notions of obligation and prohibition are logically connected, as explained in the following. Most approaches to Deontic Logic agree in assuming that, for any action A, the prohibition of A is equivalent to the obligation of omitting A:

$$\mathbf{Forb}\ A\ =\ \mathbf{Obl}\,(NON\ A) \tag{10}$$

Permissions. The third basic deontic status, besides obligations and prohibitions, is *permission*. Permissive propositions are expressed in many different ways in natural language. To express permissions in a uniform way, Deontic Logic uses the operator **Perm**. Permissions are usually represented by formulas as:

$$\mathbf{Perm}\ A$$

where A is any (positive or negative) action description, and **Perm** is the deontic operator for permission to be read as "it is permitted that".

2.2.1. Relationships between operators

The three basic deontic notions of obligation, prohibition and permission are logically connected. First of all, intuitively, when one believes that an action is obligatory, then one can conclude that the same action is permitted.

$$\mathbf{Obl}\ A\ \ \mathbf{entails}\ \ \mathbf{Perm}\ A \tag{11}$$

Since A's obligatoriness entails A's permittedness, **Obl** A is incompatible with the fact that A is not permitted:

$$\mathbf{Obl}\ A\ \ \mathbf{incompatible}\ \ \ NON\ \mathbf{Perm}\ A \tag{12}$$

The connection between the obligatoriness of A and the permittedness of A is replicated in the connection between the forbiddenness of A and the permittedness on A's omission: an action being forbidden entails permission to omit it, i.e.:

$$\mathbf{Forb}\ A\ \ \mathbf{entails}\ \ \mathbf{Perm}\ NON\ A \tag{13}$$

A being forbidden entails that the omission of A is permitted. Thus, there is a contradiction between an action being forbidden and the omission of that action not being permitted.

$$\mathbf{Forb}\ A\ \ \mathbf{incompatible}\ \ \ NON\ \mathbf{Perm}\ (NON\ A) \tag{14}$$

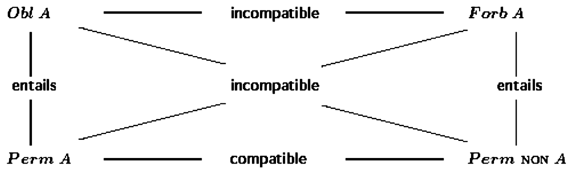
All the logical relations between deontic notions that we have just described are summarized in Fig. 1. The schema shows that there is an opposition between being obliged and being prohibited: if an action A is obligatory, then its performance is permitted, which contradicts that A is forbidden.

Similarly, if an action A is forbidden, then its omission is permitted, which contradicts that A is obligatory.

It is instead compatible that both an action A is permitted and its omission $NON\ A$ also is permitted. In such a case, A would be neither obligatory nor permitted, but *facultative*.

The deontic qualifications "obligatory" and "forbidden" are complete, in the sense that they determine the deontic status of both the action they are concerned with, and

Fig. 1 The first deontic square



the complement of that action. In fact, on the basis of the equivalence:

$$\textbf{Obl}\ \phi = \textbf{Forb}\ \textit{NON}\ \phi$$

we get the following two equivalences, the first concerning the case where ϕ is a positive action A, the second concerning the case where ϕ is the omissive action *NON A* (double negations get canceled):

$$\textbf{Obl}\ A = \textbf{Forb}\ \textit{NON}\ A \tag{15}$$

$$\textbf{Obl}\ \textit{NON}\ A = \textbf{Forb}\ A \tag{16}$$

Of course, believing that an action is permitted amounts to believing that it is not forbidden:

$$\textbf{Perm}\ A = \textit{NON}\ \textbf{Forb}\ A \tag{17}$$

This means that not being permitted amounts to being forbidden (just negate both formulas, and cancel double negations):

$$\textit{NON}\ \textbf{Perm}\ A = \textbf{Forb}\ A \tag{18}$$

From this follows that an action being permitted contradicts that action being prohibited:

$$\textbf{Perm}\ A\ \ \textbf{incompatible}\ \ \textbf{Forb}\ A \tag{19}$$

Similarly, believing that an action is obligatory amounts to excluding that its omission is permitted:

$$\textbf{Obl}\ A = \textit{NON}\ \textbf{Perm}\ \textit{NON}\ A \tag{20}$$

Correspondingly, the obligatoriness of an action (entailing the permission to perform it) contradicts the permissiveness of its omission:

$$\textbf{Obl}\ A\ \ \textbf{incompatible}\ \ \textbf{Perm}\ \textit{NON}\ A \tag{21}$$

The formulas we have just being considering are summarized in the second square of deontic notions, in Fig. 2.

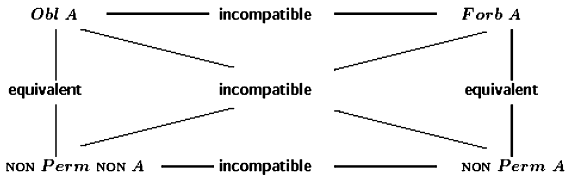


Fig. 2 The second deontic square

2.2.2. *Kripke semantics for deontic operators*

In the following, we recall the well known Kripke semantics for deontic operators.

Definition 2.9. Let P be a given set of propositional symbols.
A *frame* F is a pair $\langle W, R\rangle$ where W is a set of identifiers and $R \subseteq W \times W$.
A *model* M is pair $\langle F, V\rangle$, where $F = \langle W, R\rangle$ is a frame and $V : W \rightarrow 2^P$.

Definition 2.10. Let $M = \langle\langle W, R\rangle, V\rangle$ and $w \in W$.
$M, w \models \mathbf{Obl}(A)$ iff $\forall w'(\langle w, w'\rangle \in R \rightarrow A \in V(w))$.

In other words, an action is obligatory in a world if it is true in all the worlds accessible from it. The semantics of the other operators (prohibition, permission) is derived by that of the obligation operator:

Definition 2.11.

- $M, w \models \mathbf{Forb}(A)$ iff $M, w \models \mathbf{Obl}(\text{NON } A)$
- $M, w \models \mathbf{Perm}(A)$ iff $M, w \not\models \mathbf{Obl}(\text{NON } A)$

3. Mapping deontic operators to expectations

In this section, we propose and support intuitively and formally a mapping from deontic operators (obligation, permission, prohibition) to the expectations of the SOCS social framework.

3.1. The mapping

We propose the mapping shown in Table 1.

Table 1 Deontic notions as expectations

Operator	Abducibile
Forb *A*	**EN**(*A*)
Obl *A*	**E**(*A*)
Perm *A*	¬**EN**(*A*)
Perm *NONA*	¬**E**(*A*)

The first line of the table proposes a correspondence between the notion of prohibition (which requires an action not to be performed) and ours of negative expectation (which requires an event not to belong to the history).

In fact, the correspondence is more apparent looking at Definition 2.4, which requires, for a set of expectation to be fulfilled, the absence, in the history of events, of any event matching a negative expectation. This definition resembles closely the reduction of the prohibition operator proposed by Meyer (1988), where "it is forbidden to perform (an action) α in (a state) σ iff one performs α in σ one gets into trouble" (in that paper, "trouble" means an "undesirable state of affairs"; which is a good description of our state of violation).

Reasoning in a similar way, it is possible to notice a correspondence between the notion of obligation (which requires an action to be performed) and ours of positive expectation (which requires an event to belong to the history), as shown in the second line in Table 1.

Moreover, since a negative expectation $\mathbf{EN}(A)$ has to be read as *it is expected not* A (i.e., it is a shorthand for $\mathbf{E}(not\ A)$), its (explicit) negation, $\neg\mathbf{EN}(A)$, corresponds to permission of A. Finally, due to the logical relations among obligation, prohibition and permission discussed in Section 2.2, the fourth line of Table 1 shows how to map permission of a negative action.

3.2. The semantic link between abductive and Kripke semantics

In the following, we further support our proposed mapping by showing a link between the abductive semantics of expectations in the SOCS framework (see Section 2.1.2) and the Kripke semantics of deontic operators (see Section 2.2.2).

In order to show the link, we first give some more definition related to the SOCS framework, which correspond to concepts in the Kripke semantics.

Our corresponding of a world w is a history **HAP**, which represents some state of affairs. The corresponding of an accessible world can thus be defined as a superset of **HAP** which is compliant to the social specification.

Definition 3.1. Given a social specification $\mathcal{S}$, a history $\mathbf{HAP}'$ is a $\mathcal{S}$*-compliant extension* of a history **HAP** iff $\mathbf{HAP}' \supseteq \mathbf{HAP}$ and $\mathbf{HAP}'$ is compliant to $\mathcal{S}$.

We also single out those histories which have at least a compliant extension (corresponding to worlds which have at least one accessible world).

Definition 3.2. Given a social specification $\mathcal{S}$, a history **HAP** is *potentially* compliant to $\mathcal{S}$ if it has a $\mathcal{S}$-compliant extension.

The following definition represents the entailment of an expectation by a social specification.

Definition 3.3. An instance $\mathcal{S}_{\mathbf{HAP}}$ of a social specification $\mathcal{S}$ *requires* an expectation E iff for all **EXP** that are $\mathcal{S}_{\mathbf{HAP}}$-consistent $E \in \mathbf{EXP}$.

Table 2 Deontic and expectation concepts

Deontic	Expectation
Model	Social specification
World	History
Accessible world	Compliant extension
(deontic) entailment	requires

The following theorem is easily proven:

Theorem 3.4. *Let a history* **HAP** *be potentially compliant to a social specification* $\mathcal{S}$. *Then* ($\mathcal{S}_{\mathbf{HAP}}$ *requires* $\mathbf{E}(p)$ (*resp.* $\mathbf{EN}(p)$)) $\rightarrow$($\mathbf{H}(p)$ *is in all* (*resp. in no*) $\mathcal{S}$*-compliant extensions of* **HAP**).

Proof: Let $\mathbf{HAP}'$ be a compliant extension of **HAP**. By Definition 2.6, there exists a $\mathcal{S}_{\mathbf{HAP}'}$–consistent and $\mathbf{HAP}'$-fulfilled set **EXP** of expectations. By Definition 3.3, $\mathbf{E}(p)$ (resp. $\mathbf{EN}(p)$) is in all $\mathcal{S}_{\mathbf{HAP}}$-consistent sets of expectations, and thus also in **EXP**; but **EXP** is $\mathbf{HAP}'$-fulfilled, and thus $\mathbf{H}(p) \in$ (resp. $\notin$) $\mathbf{HAP}'$. □

Applying to Theorem 3.2 the substitutions shown in Table 2, we have one of the two implications of Definition 2.10 (the *only-if*).

The opposite, in general, does not hold in our framework. For example, consider a specification consisting of the following integrity constraint:

$$\mathbf{H}(p) \rightarrow \mathbf{EN}(q)$$

The event $\mathbf{H}(p)$ is in no compliant extension of the history $\{\mathbf{H}(q)\}$. However, $\mathbf{EN}(p)$ is not required by the social specification.

While it is possible to devise a restriction of the language which would make the reverse implication valid (social specifications composed only of disjunctions of conjunctions of expectations being possibly the simplest example), the purpose of Theorem 3.4 is to show a link between the two semantics, rather than to establish a complete equivalence.

A notable difference, from the representation point of view, is that in SOCS social integrity constraints can only express disjunctions of expectations, such that $\mathbf{E}(A) \vee \mathbf{E}(B)$ (which expresses that at least one of the two between A and B events is expected). In Deontic Logic, instead, one usually expresses the obligatoriness of disjunctions, i.e., $\mathbf{Obl}(A \vee B)$. In Kripke semantics, however, this is not equivalent to state $\mathbf{Obl}(A) \vee \mathbf{Obl}(B)$.[2]

Applying the mapping, the two integrity constraints in Spec. 2.2 can be read as follows:

1. if A requests some piece of information from B, B is either obliged to provide it by some deadline, or to refuse it, by some deadline;

[2] The two possible worlds $(A \wedge NON B)$ and $(NON A \wedge B)$ satisfy $\mathbf{Obl}(A \vee B)$, but not $\mathbf{Obl}(A) \vee \mathbf{Obl}(B)$.

2. if A provides some piece of information to B, then it is forbidden for A to refuse the same piece of information to B, at any time.

3.3. Logical relations among deontic operators as abductive integrity constraints

Let us first consider the relations summarized in the second square of deontic notions, in Fig. 2. By adopting the mapping summarized in Table 1, the equivalence relations straightforwardly arise from the uniform treatment of symbols *NON*, ¬ and *not*, and from their idempotency.

The incompatibility relations summarized in Fig. 2 emerge between the notion of obligation and prohibition (horizontal arc), and, respectively, between obligation and permission of opposite, and prohibition and non permission of opposite (diagonal arcs). By adopting the mapping summarized in Table 1, the first incompatibility is captured by the SOCS abductive semantics into the notion of E-consistency (Definition 2.3), i.e., by requiring that, for each A, the addition to the expectation set of the integrity constraint:

$$\mathbf{E}(A), \mathbf{EN}(A) \rightarrow false$$

does not lead to inconsistency.

The latter two incompatibilities (corresponding to diagonal arcs in Table 1) are captured, instead, by the notion of ¬-consistency (Definition 2.2), i.e., by requiring that, for each A, the addition to the expectation set of the integrity constraints:

$$\mathbf{E}(A), \neg\mathbf{E}(A) \rightarrow false$$

and

$$\mathbf{EN}(A), \neg\mathbf{EN}(A) \rightarrow false$$

does not lead to inconsistency.

The notions of E-consistency and ¬-consistency (and associated integrity constraints) also correspond to incompatibility relations in the first square of deontic notions, in Fig. 1.

Furthermore, the two entailment relations occurring in the first square can be captured by considering additional integrity constraints (possibly added to the set $\mathcal{IC}_S$), relating positive and negative expectations as follows:

$$\mathbf{E}(A) \rightarrow \neg\mathbf{EN}(A)$$

and

$$\mathbf{EN}(A) \rightarrow \neg\mathbf{E}(A)$$

In practice, these two constraints, when added to $\mathcal{IC}_S$ and therefore considered in $\mathcal{IC}_S$-consistency, enforce the set of expectations to be "completed", i.e., for each positive expectation $\mathbf{E}(A)$ the explicit negation of its negative counterpart, $\neg\mathbf{EN}(A)$ had to be included in the expectation set (in order to get its admissibility), and for each negative expectation $\mathbf{EN}(A)$ the explicit negation of its positive counterpart, $\neg\mathbf{E}(A)$ had to be included as well.

Finally, a notion of *regimentation* can be considered too, by enforcing obligatory actions to happen and prohibited actions not to happen. This can be easily obtained by adding to the $\mathcal{IC}_S$ the following two integrity constraints, mapping positive/negative expectations into positive/negative events:

$$\mathbf{E}(A) \rightarrow \mathbf{H}(A)$$

and

$$\mathbf{EN}(A) \rightarrow \neg\mathbf{H}(A)$$

Notice that these two conditions correspond to the (meta) integrity constraints required for fulfillment of expectation sets (see Definition 2.4). The adopted notion of fulfillment in the declarative semantics, however, just tests that these two constraints are not violated (by adopting the consistency view discussed by Fung and Kowalski, 1997), whereas if we add them to the set $\mathcal{IC}_S$, the $\mathcal{IC}_S$-consistency test (by adopting the theoremhood view, also discussed by Fung and Kowalski, 1997) would exploit them to also make events happen or not in the social environment.

4. Related work

The reduction of deontic concepts such as obligations and prohibitions has been the subject of several past works: notably, by Anderson (1958) (according to which, informally, A is obligatory iff its absence produces a state of violation) and by Meyer (1988) (where, informally, an action A is prohibited iff its being performed produces a state of violation). These two reductions strongly resemble our definition of fulfillment (Definition 2.4), which requires positive (resp. negative) expectations to have (resp. not to have) a corresponding event.

Ryu and Lee (1993) provide a first-order framework of deontic reasoning that can model and compute social regulations and norms. They employ defeasible reasoning in order to represent and manage counterfactual implications. In their framework, deontic operators are represented as first order terms; a specification is given as a set of strict and defeasible clauses. The operational semantics of their language consists of a SLD resolution-based computation process. The main purpose of our work is similar to that of Ryu and Lee's work: to give a computational method for systems specified by means of deontic operators. The works are also similar in the representation of deontic operators (as first order terms) and in the representation of the relationships among operators, such as incompatibility between obligation and prohibition (by means of rules). However, our work based on abduction, rather than on defeasible logic.

Several papers discuss "sub-ideal" situations, i.e., how to manage situations in which some of the norms are not respected.

For instance, van der Torre and Tan (1999) show the relation between diagnostic reasoning and deontic logic, importing the *principle of parsimony* from diagnostic reasoning into their deontic system, in the form of a requirement to minimize the number of violations. In particular, given the specification of a normative system (as a set of formulae which tell when a norm is violated) and a state of affairs, they define a minimal (with respect to inclusion) set of norms such that the violation of those norms is consistent with the specification and the state of affairs. The SOCS social framework, currently, only distinguishes between empty and non-empty sets of violations, and does not define minimal sets. However, it would be possible to do so by taking the minimal, with respect to inclusions, among the sets of expectations which are consistent with a social specification and a history, but possibly not fulfilled by the history. This will probably be our approach when we tackle the management of violations (by means of sanctions and recovery procedures) in future work.

Prakken and Sergot (1996) propose a solution to the problem and paradoxes stemming from earlier logical representations of *contrary-to-duty* obligations, i.e., obligations that become active when other obligations are violated. They do so by introducing a new operator $O_B(A)$, meaning that A is obligatory given the sub-ideal context B. The semantics of this operator is of Kripke type, but it differs to the standard modal logic because of the accessibility relation: in that work, the accessible worlds are the best alternatives, given the truth of B. In the "main stream" of our research, we do not support contrary-to-duty obligations. However, we proposed a modified version of our framework (Alberti et al., 2004a), which provides a simplified language and does support alternative obligations at different levels of priority; a further step could be to integrate priority levels in the main SOCS social framework.

Boella, van der Torre and Leendert (2003) discuss how a normative system can be seen as a normative agent, equipped with mental attitudes, about which other agents can reason, choosing either to fulfill their obligations, or to face the possible sanctions. Conceptually, the social infrastructure in the SOCS model could be viewed as an agent, whose knowledge base is the society specification, whose mental attitude is a set of expectations, and whose reasoning process is the $\mathcal{S}$CIFF proof procedure.

Broersen et al. (2004) investigate the deontic logic of deadlines by introducing an operator $O(\rho \leq \delta)$, which means, intuitively, that the action ρ ought to be brought about before (or at the same time) another event δ happens. They model time by means of the CTL temporal logic. We can express a similar concept by means of an integrity constraints $\mathbf{H}(\delta, T_\delta) \rightarrow \mathbf{E}(\rho, T_\rho) \wedge T_\rho \leq T_\delta$, which says that, if δ has happened, than ρ is expected to have happened before (or at the same time).

The SOCS social framework can capture, in a computational setting, the concept of (conditional) obligation with deadline presented by Dignum et al. (2002a), with an explicit mapping of time. Dignum et al. write: `Oa(r<d|p)` to state that if the precondition `p` becomes valid, the obligation becomes active. The obligation expresses the fact that `a` is expected to bring about the truth of `r` before a certain condition `d` holds.

For instance, if we have:

$$
\begin{aligned}
&p = \mathbf{H}(tell(S, a, request(G), D), T)\\
&r = \mathbf{H}(tell(a, S, answer(G), D), T'), T' > T\\
&d = T' > T + 2
\end{aligned}
$$

we can map `Oa(r<d|p)` into a IC:

$$\mathbf{H}(tell(S, a, request(G), D), T) \rightarrow$$
$$\mathbf{E}(tell(a, S, answer(G), D), T'), T' > T, T' \leq T + 2.$$

Among the organizational models, Dignum et al. (2002a, 2002b) and Dignum, Meyer and Weigand (2002c) exploit Deontic Logic to specify the society norms and rules. Their model is based on a framework which consists of three interrelated models: organizational, social and interaction. The *organizational model* defines the coordination and normative elements and describes the expected behavior of the society. Its components are roles, constraints, interaction rules, and communicative and ontology framework. The *social model* specifies the contracts that make explicit the commitments regulating the enactment of roles by individual agents. Finally, the *interaction model* describes the possible interactions between agents by specifying contracts in terms of description of agreements, rules, conditions and sanctions.

Deontic operators have been used not only at the social level, but also at the agent level. Notably, in IMPACT (Arisha et al., 1999; Eiter, Subrahmanian and Pick, 1999), agent programs may be used to specify what an agent is obliged to do, what an agent may do, and what an agent cannot do on the basis of deontic operators of Permission, Obligation and Prohibition (whose semantics does not rely on a Deontic Logic semantics). In this respect, the IMPACT and SOCS social models have similarities even if their purpose and expressivity are different. The main difference is that the goal of agent programs in IMPACT is to express and determine by its application the behavior of a single agent, whereas the SOCS social model goal is to express rules of interaction and norms, that instead cannot really determine and constrain the behavior of the single agents participating to a society, since agents are autonomous.

5. Conclusions and future work

In this paper, we propose a mapping of the usual deontic operators (obligation, prohibitions, permission) to a particular kind of abducible predicates (expectations) introduced in the SOCS social framework. We support the mapping by showing the similarity of the concepts at an intuitive and at a formal level.

The mapping makes it possible to exploit the operational counterpart of the SOCS social framework, the $\mathcal{S}$CIFF abductive proof procedure, to verify the compliance of agent behaviour to a specification given in terms of deontic operators.

Future work will be devoted to the actual implementation of deontic specifications using our framework. This may require an application of our language to more sophisticated social and normative models (such as the one recently proposed by López y López et al. (2005)), and possibly a reconsideration some of the aspects of the language, especially in order to support exceptions and contrary to duty obligations.

Acknowledgments This work has been supported by the European Commission within the SOCS project (IST-2001-32530), funded within the Global Computing Programme and by the MIUR COFIN 2003 projects *La Gestione e la negoziazione automatica dei diritti sulle opere dell'ingegno digitali: aspetti giuridici e informatici* and *Sviluppo e verifica di sistemi multiagente basati sulla logica.*

References

Alberti M, Gavanelli M, Lamma E, Mello P, Torroni P (2003a) An abductive interpretation for open societies. In: Cappelli A, Turini F (eds) AI*IA 2003: Advances in artificial intelligence, Proceedings of the 8th Congress of the Italian association for artificial intelligence, Pisa (Lecture Notes in Artificial Intelligence), vol. 2829. Springer-Verlag, pp 287–299

Alberti M, Ciampolini A, Gavanelli M, Lamma E, Mello P, Torroni, P (2003b) A social ACL semantics by deontic constraints. In: Mařík V, Müller J, Pěchouček M (eds) Multi-agent systems and applications III. Proceedings of the 3rd international central and eastern european conference on multi-agent systems, CEEMAS 2003. (Lecture Notes in Artificial Intelligence), vol. 2691. Springer-Verlag, Prague, Czech Republic: pp 204–213

Alberti M, Gavanelli M, Lamma E, Mello P, Torroni, P (2003c) Specification and verification of agent interactions using social integrity constraints. Electr Notes Theor Comp Sci 85(2)

Alberti M, Daolio D, Gavanelli M, Lamma E, Mello P, Torroni P (2004a) Specification and verification of agent interaction protocols in a logic-based system. In: Haddad H M, Omicini A, Wainwright R L (eds) Proceedings of the 19th Annual ACM Symposium on Applied Computing (SAC 2004). Special Track on Agents, Interactions, Mobility, and Systems (AIMS). ACM Press, Nicosia, Cyprus pp 72–78

Alberti M, Chesani F, Gavanelli M, Lamma E, Mello P, Torroni P (2005a) The SOCS computational logic approach for the specification and verification of agent societies. In: Priami C, Quaglia P (eds) Global Computing: IST/FET International Workshop, GC 2004 Rovereto, Italy, March 9–12, 2004 Revised Selected Papers. (Lecture Notes in Artificial Intelligence), vol. 3267. Springer-Verlag, pp 324–339

Alberti M, Gavanelli M, Lamma E, Mello P, Torroni P (2005b) The SCIFF abductive proof procedure. In Bandini S, Manzoni S (eds) Proceedings of the 9th National Congress on Artificial Intelligence, AI*IA 2005. (Lecture Notes in Artificial Intelligence) vol. 3673. Springer-Verlag, Milan, Italy, pp 135–147

Alberti M, Chesani F, Gavanelli M, Lamma E, Mello P, Torroni P (2006) Compliance verification of agent interaction: a logic-based tool. Applied Artificial Intelligence 20(2–4):133–157

ALFEBIITE (1999) ALFEBIITE: A logical framework for ethical behaviour between infohabitants in the information trading economy of the universal information ecosystem. IST-1999-10298. http://www.iis.ee.ic.ac.uk/ alfebiite/ab-home.htm

Anderson A (1958) A reduction of deontic logic to alethic modal logic. Mind 67:100–103

Apt KR, Bol RN (1994) Logic programming and negation: A survey. J Logic Progr 19/20:9–71

Arisha KA, Ozcan F, Ross R, Subrahmanian VS, Eiter T, Kraus S (1999) IMPACT: A platform for collaborating agents. IEEE Intell Syst 14(2):64–72

Artikis A, Pitt J, Sergot M (2002) Animated specifications of computational societies. In: Castelfranchi C, Lewis Johnson W (eds) Proceedings of the first international joint conference on autonomous agents and multiagent systems (AAMAS-2002), Part III. ACM Press Bologna, Italy pp 1053–1061

Boella G, van der Torre, L WN (2003) Attributing mental attitudes to normative systems. In: Rosenschein JS, Sandholm T, Wooldridge M, Yokoo M (eds) Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2003). ACM Press Melbourne, Victoria pp 942–943

Boella G, van der Torre L, Verhagen H (eds) (2005) Proceedings of the symposium on normative multi agent-systems. University of Hertfordshire, Hatfield, UK

Bracciali A, Demetriou N, Endriss U, Kakas A, Lu W, Mancarella P, Sadri F, Stathis K, Toni F, Terreni G (2005) The KGP model of agency: Computational model and prototype implementation. In: Priami C, Quaglia P (eds) Global computing: IST/FET international workshop, GC 2004 Rovereto, Italy, March 9-12, 2004 Revised Selected Papers (Lecture Notes in Artificial Intelligence), vol. 3267. Springer-Verlag, pp 340–367

Broersen J, Dignum F, Dignum V, Meyer J-J Ch (2004) Designing a deontic logic of deadlines. In: Lomuscio A, Nute D (eds) DEON (Lecture Notes in Computer Science), vol. 3065. Springer, pp 43–56

Castelfranchi C, Dignum F, Jonker CM, Treur J (1999) Deliberative normative agents: Principles and architecture. In: Jennings NR, Lespérance Y (eds) Intelligent agents VI, Agent theories, architectures, and languages, 6th International Workshop, ATAL '99, Orlando, Florida, USA, Proceedings. (Lecture Notes in Computer Science), no. 1757. Springer-Verlag, pp 364–378

Clark KL (1978) Negation as failure. In: Gallaire H, Minker J (eds) Logic and data bases. Plenum Press, pp 293–322

Conte R, Falcone R, Sartor G (1999) Special issue on agents and norms. Art Intell Law 1(7)

Dignum V, Meyer JJ, Dignum F, Weigand H (2002a) Formal specification of interaction in agent societies. In: Proceedings of the second goddard workshop on formal approaches to agent-based systems (FAABS), Maryland
Dignum V, Meyer JJ, Weigand H, Dignum F (2002b) An organizational-oriented model for agent societies. In: Proceedings of International Workshop on Regulated Agent-Based Social Systems: Theories and Applications. AAMAS'02, Bologna
Dignum V, Meyer JJ, Weigand H (2002c) Towards an organizational model for agent societies using contracts. In: Castelfranchi C, Lewis Johnson W (eds) Proceedings of the first international joint conference on autonomous agents and multiagent systems (AAMAS-2002), Part II. Bologna, Italy: ACM Press, pp 694–695
Eiter T, Subrahmanian VS, Pick G (1999) Heterogeneous active agents, I: Semantics. Art Intell 108(1–2):179–255
Esteva M, de la Cruz D, Sierra C (2002) ISLANDER: An electronic institutions editor. In: Castelfranchi C, Lewis Johnson W (eds) Proceedings of the first international joint conference on autonomous agents and multiagent systems (AAMAS-2002), Part III. ACM Press, Bologna, Italy pp 1045–1052
Fung TH, Kowalski RA (1997) The IFF proof procedure for abductive logic programming. J Logic Progr 33(2): 151–165
Hewitt C (1991) Open information systems semantics for distributed art intell. Art Intell 47(1–3):79–106
Kakas AC, Kowalski RA, Toni F (1993) Abductive logic programming. J Logic Comput 2(6):719–770
Kunen K (1987) Negation in logic progr J Logic Progr vol. 4, pp 289–308
López y López, Fabiola, Luck, Michael, d'Inverno, Mark (2005) A normative framework for agent-based systems. In: Boella et al., (2005)
Meyer JJ Ch (1988) A different approach to deontic logic: Deontic logic viewed as a variant of dynamic logic. Notre Dame J. Form Logic 29(1):109–136
Noriega P, Sierra C (2002) Institutions in perspective: An extended abstract. In: 6th International Workshop CIA-2002 on Cooperative Information Agents (Lecture Notes in Artificial Intelligence), vol. 2446. Springer-Verlag
Prakken H, Sergot M (1996) Contrary-to-duty obligations. Studia Logica 57(1):91–115
Ryu Young U, Lee Ronald M (1993) Defeasible deontic reasoning: A logic programming model. In: Meyer J-J Ch, Wieringa RJ (eds) Deontic logic in computer science: normative system specification. John Wiley & Sons Ltd, pp 225–241
Sadri F, Stathis K, Toni F (2005) Normative KGP agents: A preliminary report. In: Boella et al. (2005)
Sartor G (2004) Legal reasoning. Treatise, vol. 5. Kluwer Dordrecht
SOCS (2002) Societies of computeeS (SOCS): A computational logic model for the description, analysis and verification of global and open societies of heterogeneous computees. IST-2001-32530. Home Page: http://lia.deis.unibo.it/Research/SOCS/
van der Torre L (2003) Contextual deontic logic: Normative agents, violations and independence. Ann Math Art Intell 37(1):33–63
van der Torre LWN, Tan Y-H (1999) Diagnosis and decision making in normative reasoning. Artif. Intell. Law 7(1):51–67
Wright GH (1951) Deontic logic. Mind 60:1–15

Marco Alberti received his laurea degree in Electronic Engineering in 2001 and his Ph.D. in Information Engineering in 2005 from the University of Ferrara, Italy. His research interests include constraint logic programming and abductive logic programming, applied in particular to the specification and verification of multi-agent systems. He has been involved as a research assistants in national and European research projects. He currently has a post-doc position in the Department of Engineering at the University of Ferrara.

Marco Gavanelli is currently assistant professor in the Department of Engineering at the University of Ferrara, Italy. He graduated in Computer Science Engineering in 1998 at the University of Bologna, Italy. He got his Ph.D. in 2002 at Ferrara University. His research interest include Artificial Intelligence, Constraint Logic Programming, Multi-criteria Optimisation, Abductive Logic Programming, Multi-Agent Systems. He is a member of ALP (the Association for Logic Programming) and AI*IA (the Italian Association for Artificial Intelligence). He has organised workshops, and is author of more than 30 publications between journals and conference proceedings.

Evelina Lamma received her degree in Electronic Engineering from University of Bologna, Italy, in 1985 and her Ph.D. degree in Computer Science in 1990. Currently she is Full Professor at the Faculty of Engineering of the University of Ferrara where she teaches Artificial Intelligence and Foundations of Computer Science. Her research activity focuses around:

– programming languages (logic languages, modular and object-oriented programming);
– artificial intelligence;
– knowledge representation;
– intelligent agents and multi-agent systems;
– machine learning.

Her research has covered implementation, application and theoretical aspects. She took part to several national and international research projects. She was responsible of the research group at the Dipartimento di Ingegneria of the University of Ferrara in the UE ITS-2001-32530 Project (named SOCS), in the the context of the UE V Framework Programme - Global Computing Action.

Paola Mello received her degree in Electronic Engineering from the University of Bologna, Italy, in 1982, and her Ph.D. degree in Computer Science in 1989. Since 1994 she has been Full Professor. She is enrolled, at present, at the Faculty of Engineering of the University of Bologna (Italy), where she teaches Artificial Intelligence. Her research activity focuses on programming languages, with particular reference to logic languages and their extensions, artificial intelligence, knowledge representation, expert systems with particular emphasis on medical applications, and multi-agent systems. Her research has covered implementation, application and theoretical aspects and is presented in several national and international publications. She took part to several national and international research projects in the context of computational logic.

Giovanni Sartor is Marie-Curie professor of Legal informatics and Legal Theory at the European University Institute of Florence and professor of Computer and Law at the University of Bologna (on leave), after obtaining a PhD at the European University Institute (Florence), working at the Court of Justice of the European Union (Luxembourg), being a researcher at the Italian National Council of Research (ITTIG, Florence), and holding the chair in Jurisprudence at Queen's University of Belfast (where he now is honorary professor). He is co-editor of the Artificial Intelligence and Law Journal and has published widely in legal philosophy, computational logic, legislation technique, and computer law.

Paolo Torroni is Assistant Professor in computing at the Faculty of Engineering of the University of Bologna, Italy. He obtained a PhD in Computer Science and Electronic Engineering in 2002, with a dissertation on logic-based agent reasoning and interaction. His research interests mainly focus on computational logic and multi-agent systems research, including logic programming, abductive and hypothetical reasoning, agent interaction, dialogue, negotiation, and argumentation. He is in the steering committee of the CLIMA and DALT international workshops and of the Italian logic programming interest group GULP.