

Abduction and Legal Reasoning

Giada Maggenti
Dipartimento di Informatica e
Sistemistica
Università di Pavia
Via Ferrata, 1 – 27100 Pavia,
Italy
giada.maggenti@unipv.it

Andrea Bracciali
Dipartimento di Informatica
Università di Pisa
Largo B. Pontecorvo, 3 –
56127 – Pisa, Italy
braccia@di.unipi.it

Paolo Mancarella
Dipartimento di Informatica
Università di Pisa
Largo B. Pontecorvo, 3 –
56127 – Pisa, Italy
paolo@di.unipi.it

ABSTRACT

In this paper we present LAILA⁺, an extension of the LAILA coordination language for abductive logic agents, i.e. reasoning agents that collaborate towards the solution of a given problem exploiting a set of distributed, possibly partial, knowledge of the application domain. The extension consists of *i*) the possibility for agents to communicate with each other hypotheses while devising a coordinated solution, and *ii*) a relaxed consistency mechanism based on a given agent hierarchy: stronger agent coherence may overcome weaker agent inconsistency. We argue that the framework well adapts to legal reasoning, with agents that try to prove/disprove evidences from different, possibly partial and hierarchical, viewpoints, as often happens for instance in a trial.

1. INTRODUCTION

We present a multi-agent framework based on abductive reasoning for modeling a distributed knowledge that may result in being incomplete and globally not consistent. Our starting point is the LAILA coordination model [1], according to which agents interact in order to provide solutions for given problems. Each agent is equipped with a different logical-abductive program, by means of which it devises solutions that may be based on (abductive) hypotheses. These are required to be globally consistent within the pool of agents. We extend the LAILA model towards two main directions: while interacting agents may not only ask the collaboration of other agents to solve their goals, but also they can explicitly communicate and negotiate the assumed hypotheses, and, moreover, hypotheses may not be globally consistent, but only locally and according to a given hierarchy amongst agents. Such extensions define a new coordination language, that we call LAILA⁺, which appears to be more expressive than the original one.

The newly proposed framework well resembles forensic reasoning, that can be abstracted as the activity of a pool of

independent agents, possibly with partial and not necessarily consistent views of a problem, cooperating or competing in order to determine a solution, as might happen, say, while discussing a case in court. Moreover, the pool may be hierarchically organized. We will stress the suitability of the framework for legal reasoning by applying it to an abstraction of the development of a trial, already appeared in literature. The trial comprises a judge and two opponent lawyers.

Next, a brief account about the relations between computer science and the legal environment is given, together with some background about abductive reasoning and multi-agent systems. In the next two sections, LAILA is recapped and its extension LAILA⁺ is introduced. Then a case study in the legal contest is discussed, illustrating the main aspects, and finally, some concluding remarks are drawn.

1.1 Law and Computer Science

The interaction between law and computer science can take place in different ways:

- *Computer science object of law*, that concerns, for example, information contracts, efficacy of digital signature, protection of software;
- *law object of computer science*, that is the science engaged into conception and production of applications with legal interest or legal content.

In this latter case we talk about *legal informatics*. Several functional application typology can be found within legal informatics, like *legal informatics with documentary orientation*, *legal informatics with managerial orientation*; and *legal informatics with consulting and decision-making orientation*. In general, the existing applications in legal informatics can be grouped in four main classes [3, 7]: *i) systems for legal analysis*: they are used for establish legal qualification of specific situations. They accept, as the input, the description of a case and give, as the output, a classification of the case; *ii) system for legal planning*: they suggest action plans into legal contexts, taking the objectives and the encumbrances into consideration; *iii) system for conceptual research of legal information (conceptual information retrieval)*; *iv) system for intelligence write of legal documents*: they assist the jurist into preparation of texts.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

e-Forensics 2008, January 21-23, 2008, Adelaide, Australia.
© 2008 ICST 978-963-9799-19-6.

Considerable interest has been raised by legal informatics since the end of the '60s and the begin of the '70s, many people believing that the methodology of computer science and cybernetics could have revolutionized legal studies. In those days, however, the contribution of computer science has been quite modest and so enthusiasm changed itself into disappointment and the development of the field had a slow down [3, 7]. At the end of the '80s, some informatics systems for applying standard legal reasoning have been developed, examples exist for instance in the taxation context. Moreover, the Artificial Intelligence (AI) research area has identified in law an ideal field of analysis, checking and experimentation, and several approaches developing formal methods for processing uncertain and contrasting data have been developed.

1.2 Abductive Reasoning

We follow the well-known classification of reasoning, understood as the process by means of which thought and the capability of making logical inferences is expressed, given by Pierce in [4] (where firstly abduction has been accounted for):

- *deduction*: it is an analytical process based on application of general rules to particular cases, with the inference of a result;
- *induction*: it is a kind of concise reasoning that infers general rules from particular cases;
- *abduction*: it is a kind of concise reasoning that infers particular cases from general observations and rules; this is a weak kind of inference because we cannot say that the explanation *is* true, but that it *can be* true.

Abduction can be seen as the opposite of deduction, in fact while deduction makes conclusions from a set of given preconditions, abduction, given: *i*) an assert to explain or justify; *ii*) a set of known events and rules and a set of constraints to respect; *iii*) a set of possible hypothesis, identifies groups of compatible hypotheses, which are compatible with events, rules and constraints, and are sufficient to explain the given assert. This is a powerful hypothetic reasoning mechanism, with incomplete knowledge. Let us informally consider the following example. Given the rules:

grass_is_wet *if* rained_last_night.

grass_is_wet *if* sprinkler_was_on.

shoes_are_wet *if* grass_is_wet.

and the observation that our shoes are wet, {rained_last_night} is a possible explanation. Note that this fact has not an explanation itself and it must then be considered as a hypothesis. Observe that also {sprinkler_was_on} is an explaining hypothesis, and that, assuming that the sprinkler does not operate while raining (constraint: rained_last_night and sprinkler_was_on \rightarrow false), it would not be consistent together with the hypothesis on rain.

1.2.1 Definitions

Some basic definitions are recapped below (see, e.g., [1]).

Definition 1. A *logic program* is a set of clauses $A \leftarrow L_1, \dots, L_n$, where A is an atom (i.e. a predicate, $p(t_1, \dots, t_n)$, where t_i is either a constant, a variable or a compound term) and each L_i is a literal (i.e. an atom or the negation of an atom). A is the *head* of the clause and L_1, \dots, L_n is the *body*.

Definition 2. An *abductive logic program* is a triple $\langle P, A, IC \rangle$ where P is a logic program, A is a set of predicates, called *abducibles* and IC is a set of sentences, referred to as *integrity constraints*.

Without loss of generality, A is a subset of undefined predicates in P . Abducibles can be used to extend P in order to explain given sentences, called *queries*, or *observations* or *goals*. That is, *abduction* is the process of determining a set $\Delta \subseteq A$ of abducibles, such that¹:

$$T \cup \Delta \models G, \text{ and}$$

$$T \cup IC \cup \Delta \not\models \perp,$$

where T is a theory and G is an observation to demonstrate.

If such a Δ exists, it is called *abductive explanation* for G , denoted by:

$$\langle T, A, IC \rangle \models_{abd}^{\Delta} G$$

In this context, IC determines the set of assumptions that can coherently be made together.

1.3 Agents and Multi-Agent Systems

Multi-Agents systems are composed by autonomous computational entities, i.e. agents, which interact, possibly cooperatively, in order to address a given issue in a given environment. Protocols and normative prescriptions may rule the way they communicate and interact, each one however having its own independent control. Literature about Multi-Agent systems is huge, for further details we refer the interested reader to any recognized proposal or surveying work ([9, 8]).

2. THE LAILA LANGUAGE

The coordination language LAILA (Language for Abductive Logic Agents) has been defined in [2, 1] with the aim of modeling the interaction amongst logic agents able to reason abductively. The agents can collaborate or compete for the solution of problems and their social behaviour, i.e. distributed reasoning, can be easily expressed within the LAILA framework.

By means of the collaboration operator ($\&$) and the competition operator ($;$) each agent can formulate queries to a set of agents and coordinate the way they solve them. This is done by means of the basic communication operator ($>$), used to “distribute” queries from an agent to the others, and the “down-reflection” operator (\downarrow), used to fire an abductive

¹For simplicity, T is assumed ground and stratified. Then, standard semantics, like stable model and well-founded, coincide and \models stands for any of them.

derivation, local to a single agent. A LAILA program is a set of clauses defined as follows [2]:

L-Clause ::= Atom :- L-Body.

L-Body ::= L-Expr , L-Body | L-Expr

L-Expr ::= L-Fact & L-Expr | L-Fact ; L-Expr | L-Fact

L-Fact ::= \downarrow LList | Agent $>$ LList | (L-Body) | LList

where LList is a list of literals. A computation starts with a query: Query ::= ? L-Body. For instance,

? \downarrow (p(a) & A2 $>$ (q(b)).

executed by agent A1 stands for A1 proving p(a) and asking agent A2 to prove q(b). Cooperation and consistency amongst agent is made precise according to the following semantics.

2.1 Semantic Rules of LAILA

By $A \vdash_{\delta}^B q$ we indicate that the agent A , collaborating with a set of agents B , is able to prove the query q , assuming the set of hypotheses (abducibles) δ . the above judgment is proved according to rules like the ones proposed in [2]:

Down-reflection Formula:

$$\frac{A \vdash_{\delta}^{abd} G}{A \vdash_{\delta}^A \downarrow G}$$

If the agent A can autonomously prove the goal G with assumptions δ , then the same assumptions prove the down-reflected query $\downarrow G$ at the level of the coordination language.

Communication Formula:

$$\frac{Y \vdash_{\delta_1}^B G \wedge B \cup X \vdash_{\delta}^{cons} \delta_1}{X \vdash_{\delta}^{B \cup X} Y > G}$$

If the agent Y can prove G together with agents in B (possibly asked by Y), and the assumptions made δ_1 are consistent (no constraint violated) for *all* the agents involved including agent X (note that extra assumptions δ might be required), then X can prove G by asking Y to do this.

Competitive Formula:

$$\frac{A \vdash_{\delta}^B f}{A \vdash_{\delta}^B f; F} \quad \frac{A \vdash_{\delta}^B F}{A \vdash_{\delta}^B f; F}$$

If A can prove a (distributed) query in a competitive (;) list, then the list itself is considered as proved (notice that this introduces a non-deterministic choice of one of the rules introduced above).

Collaborative Formula:

$$\frac{A \vdash_{\delta'}^{B'} f \wedge A \vdash_{\delta''}^{B''} F \wedge B' \cup B'' \vdash_{\delta}^{cons} \delta' \cup \delta''}{A \vdash_{\delta}^{B' \cup B''} f \& F}$$

Like above, but all the queries in the list must be proved by means of an overall consistent set of hypotheses.

Goal Formula: A suitable set of rules defines the abductive reasoning capabilities of each agent in a standard way (see [2]).

Consistency Formula

$$\frac{\forall A_i \in B. A_i \vdash_{\delta_i}^{abd} \delta \wedge \delta \subset \bigcup_{A_j \in B} \delta_j \wedge B \vdash_{\delta'}^{cons} \bigcup_{A_i \in B} \delta_i}{B \vdash_{\delta'}^{cons} \delta}$$

$$\frac{\forall A_i \in B. A_i \vdash_{\delta}^{abd} \delta}{B \vdash_{\delta}^{cons} \delta}$$

Finally, rules are given to define global consistency in terms of (global) fulfillment of the integrity constraints (2^{nd} rule: each agent proves the hypotheses) of all the agents involved (1^{st} rule: each agent proves hypotheses, possibly adding new ones, then checked by other agents).

The LAILA framework has been implemented in the ALIAS (**A**bductive **L**og**I**c **A**gent **S**ystem), which, based on Java, offers the distributed communication infrastructure to LAILA agents.

3. THE LAILA+ LANGUAGE

We have extended LAILA towards two main directions:

1. consistency is not required to be global always, but two extra kinds of consistency (vertical and horizontal) are possible, and
2. sets of hypotheses can be explicitly communicated to agents, requiring agents to consider them in their proofs.

3.1 LAILA+ Syntax

These extensions have required changes in the syntax of the language. LAILA+ now features three communications:

1. the *communication operator 1:1*: a sender agent asks another (set of) agent(s) to prove a query, according to a chosen kind of consistency ($>_O$ or $>_V$);
2. the *collaborative communication operator*: a sender agent asks a set of agents to prove a query. All the agents involved must prove their queries, according to the chosen kind of consistency ($>_O \dots \& \dots$, or $>_V \dots \& \dots$);
3. the *competitive communication operator*: a sender agent asks a set of agents to prove a query. At least one of the agents involved must prove the assigned query, according to the chosen kind of consistency ($>_O \dots || \dots$).

Each agent is now defined as $\langle Rc, T, A, IC \rangle$, where Rc is a set of clauses for communicating with other agents. These clauses have form: $Atom \leftarrow Body$, where:

Body ::= Literal | \downarrow Literal | Literal, Body | \downarrow Literal, Body
| Comm | Comm , Body
Comm ::= $A >_O$ Body & Body | $A >_O$ Body || Body
| $A >_V$ Body
Body ::= B ($\{\Delta_{in}\}$, Goal) | Body & Body |
Body || Body

where $\{\Delta_{in}\}$ is a set abducibles sent to the receiver agent (different from LAILA), which is required to be different from the sender. The components T , A and IC are standard for abductive theories.

A computation starts with a query: Query ::= ?Goal, relative to a given agent. Again, the judgment $A \vdash_{\delta}^B q$, defined below, is read as agent A , collaborating with a set of agents B , is able to prove the query q , assuming the set of hypotheses (abducibles) δ .

The differences between “ $>_O$ ” and “ $>_V$ ” and their application to legal informatics are informally discussed in the next paragraph.

3.2 Consistency and Hierarchy

Agents can be organized in a hierarchy way, reflecting for instance distribution of roles. In the legal environment it easy to recognise a hierarchy amongst different classes of judges and lawyers, but also between different class of sentences. In these setting the two kind of consistency introduced have to be read as:

- *horizontal consistency*: it is used for the collaborative communication: all the agents involved in proving a query required by a sender agent must agree, together with the sender, on the assumed hypotheses;
- *vertical consistency*: it is used in all three types of communication: in the 1:1 communication there are only 2 agents that must be consistent on the hypotheses they assume; in the competitive communication it is instead required that the sender agent and the one which provides the proof are consistent; in the collaborative communication all the receiver agents must be consistent with the sender one, but it is not required that all the receivers are consistent one with another (different from LAILA).

3.3 Semantic Rules

Goal Formula:

$$\frac{\exists h \leftarrow C \in Rc_A, A \vdash_{\delta}^B C}{A \vdash_{\delta}^B h}$$

$$\frac{A \vdash_{\delta_1}^{B_1} h, A \vdash_{\delta_2}^{B_2} C, A \cup B_1 \cup B_2 \vdash_{\delta}^{cons} \delta_1 \cup \delta_2}{A \vdash_{\delta}^{B_1 \cup B_2} h, C}$$

In the first rule, to do the demonstration, the agent A searches among its rules, one which the head matched with that it must demonstrate. After this, the agent A must demonstrate the body of the rule. The second rule is used to composition of predicates, when that is the body is composed by more parts. Unlike the corresponded LAILA rule, here is required a consistency check among the agents that to the computation/demonstration.

Down-reflection Formula:

$$\frac{A \vdash_{\delta}^{abd} G}{A \vdash_{\delta}^A \downarrow G}$$

$$\frac{A \vdash_{\delta}^{abd} \{\Delta_{in}\}}{A \vdash_{\delta}^A \downarrow \{\Delta_{in}\}}$$

In the first rule, the agent A demonstrates, with an abductive procedure, the goal G thanks to its own KB. The second one is not present in the original language: in this rule, the set $\{\Delta_{in}\}$ is a set made of abducibles that the agent A demonstrates thanks to its local KB.

1:1 Communication Formula:

$$\frac{A_1 \vdash_{\delta_1}^B \downarrow \{\Delta_{in}\}, G \quad A_0 \cup B \vdash_{\delta}^{cons} \delta_1}{A_0 \vdash_{\delta}^{A_0 \cup B} A_0 >_V A_1(\{\Delta_{in}\}, G)}$$

where A_0 is the sender and A_1 is the receiver and are different. The consistency is vertical: A_0 is consistent with the other agents of the set B . The set Δ_{in} is contained in the output set δ , since, in order for the agents to be consistent, more hypotheses may need to be assumed. The difference between the LAILA-rule is the presence of the set Δ_{in} that the sender passes on to the receiver so that the latter can use it to demonstrate its goal.

Collaborative Communication Formula:

- with horizontal consistency:

$$\frac{A_1 \vdash_{\delta_1}^{B_1} \downarrow \{\Delta_{1-in}\}, G_1 \dots A_n \vdash_{\delta_n}^{B_n} \downarrow \{\Delta_{n-in}\}, G_n}{A_0 \cup \mathbf{B} \vdash_{\delta}^{cons} \delta_1 \cup \dots \cup \delta_n} \\ A_0 \vdash_{\delta}^{A_0 \cup \mathbf{B}} A_0 >_O A_1(\{\Delta_{1-in}\}, G_1) \& \dots \& A_n(\{\Delta_{n-in}\}, G_n)$$

where $\mathbf{B} = B_1 \cup \dots \cup B_n$. Note that, as in the previous rule, $\delta_i \supseteq \Delta_{i-in}$. With this type of consistency, all the agents involved must be consistent all together and with the sender agent A_0 . The output is δ . G_i and Δ_{i-in} may differ varying i . The sender is required to be different from the receivers.

- with vertical consistency:

$$\frac{A_1 \vdash_{\delta_1}^{B_1} \downarrow \{\Delta_{1-in}\}, G_1 \dots A_n \vdash_{\delta_n}^{B_n} \downarrow \{\Delta_{n-in}\}, G_n}{A_0 \cup B_1 \vdash_{\delta_1}^{cons} \delta_1, \dots, A_0 \cup B_n \vdash_{\delta_n}^{cons} \delta_n} \\ A_0 \vdash_{\delta}^{A_0 \cup \mathbf{B}} A_0 >_V A_1(\{\Delta_{1-in}\}, G_1) \& \dots \& A_n(\{\Delta_{n-in}\}, G_n)$$

where $\mathbf{B} = B_1 \cup \dots \cup B_n$. With this type of consistency, every agent involved must be consistent with the sender agent only. Like in the previous rule, $\delta_i \supseteq \Delta_{i-in}$, G_i and Δ_{i-in} must not be all the same, and the sender must be different from the receivers.

Competitive Communication Formula:

$$\frac{\exists i \in [1, n]. A_i \vdash_{\delta_i}^{B_i} \downarrow \{\Delta_{i-in}\}, G_i \quad A_0 \cup B_i \vdash_{\delta}^{cons} \delta_i}{A_0 \vdash_{\delta}^{A_0 \cup B_1} A_0 >_V A_1(\{\Delta_{1-in}\}, G_1) \parallel \dots \parallel A_n(\{\Delta_{n-in}\}, G_n)}$$

In this rule, the consistency is only vertical: the receiver agent that demonstrates G_i must be consistent with the sender agent. As in the previous rules G_i 's and Δ_{i-in} 's can be different, the sender must be different from the receivers and $\delta_i \supseteq \Delta_{i-in}$.

Consistency Formula:

$$\frac{\forall A_i \in B. A_i \vdash_{\delta}^{abd} \delta}{B \vdash_{\delta}^{cons} \delta}$$

$$\frac{\forall A_i \in B. A_i \vdash_{\delta_i}^{abd} \delta, \delta \subset \bigcup_{A_j \in B} \delta_j, B \vdash_{\delta^*}^{cons} \bigcup_{A_j \in B} \delta_j}{B \vdash_{\delta^*}^{cons} \delta}$$

In the first rule, no abducibles are added to the set δ . In the second rule, given a set δ of abducibles, consistent with each agent in B , more abducibles can be added to it by any of the agents involved, provided that the new set is consistent. When no more abducibles need to be added the process terminates applying the first rule and the final set of abducibles is given as output.

4. LEGAL REASONING IN LAILA⁺

We present an example of how our framework adapts to legal reasoning. The example is taken from the literature [6] where an “argument-based” system is used, i.e. the contrast of different, “defeasible”, arguments in favor or contrary to the demonstration of a given fact. This takes the shape of a dialogue between a proponent and an opponent of an argument [5]. This dialogue is based on the fact that every assumption or conclusion of an argument made by a party can be attacked by the other party. For space limitation, we present the example, i.e. a trial transcript, and illustrate how the defined constructs apply and, informally, which is the outcome of the proof (the case study has been developed into details in [3]).

Knowledge: the body of Bill has been found; Bill has been killed. The crime weapon is a knife. John is guilty, because his fingerprints were on the knife. There exists an anonymous witness, X, that has seen John on the crime scene while he pulled the knife out from the dead body of Bill and this could explain John’s fingerprints on the knife.

We indicate with P the proponent and with O the opponent, and so the legal dispute is the following [6]:

P : I claim that John is guilty of murder.

O : I deny your claim.

P : John’s fingerprints were on the knife. If someone stabs a person to death, his fingerprints must be on the knife, so John has stabbed Bill to death. If a person stabs someone to death, he is guilty of murder, so John is guilty of murder.

O : I concede your premises, but I disagree that they imply your claim: witness X says that John had pulled the knife out of the dead body. This explains why fingerprints were on the knife.

P : X ’s testimony is inadmissible evidence, since she is anonymous. Therefore, my claim stands.

In this example there are three agents:

- the Judge_Agent, denoted by JA,
- the Charge_Lawyer_Agent, denoted by CLA, and
- the Defence_Lawyer_Agent, denoted by DLA.

With our formalization, the computation starts with the following query, made to JA.

$?guilty(John, \{\})$.

JA has, among its rules, a communication one between itself and CLA to demonstrate that John is guilty:

$guilty(c, \{\Delta_{in}\}) \leftarrow JA >_V CLA(\{\Delta_{in}\}, guilty(c))$.

The CLA is able to demonstrate this, assuming that the crime weapon is a knife and that Bill is killed by John. Let $\delta_1 = \{crime_weapon(knife), killed(Bill, John)\}$ be such a set of hypotheses.

Then, the JA considers again John’s guilty under the set of hypotheses δ_1 :

$?guilty(John, \delta_1)$

The JA has, among its rules, a communication one between itself and the DLA to demonstrate the innocence of John:

$notguilty(c, \{\Delta_{in}\}) \leftarrow JA >_V DLA(\{\Delta_{in}\}, notguilty(c))$.

The DLA agent is able to demonstrate this, assuming the existence of a witness along with the set of hypotheses δ_1 . Let δ_2 be the extended set of hypotheses.

At this point, the JA asks to the CLA to demonstrate that John is guilty, passing $\Delta_{in} = \delta_2$, and the lawyer agent demonstrates it, assuming that witness’s testimony is inadmissible, since she is anonymous. So, the JA can pass guilt sentence against John (the complete formalization of the example can be found in [3]).

It is worth noticing that this conclusion has been proved by requiring consistency only between the judge and each lawyer separately (notice $>_V$ in all communications), and that the knowledge of each lawyer separately could lead to different conclusions.

5. CONCLUSIONS AND FUTURE WORKS

A single type of (global) consistency as in the original LAILA language does not fit the presented example and motivates the extension we proposed in LAILA⁺. Due to the vertical consistency, the Defence_Lawyer_Agent and the Judge_Agent are consistent with one another, the Charge_Lawyer_Agent and the Judge_Agent are consistent with one another, while it is not required that the two lawyer agents are consistent all together. We believe this paradigmatic of legal reasoning, and hence the approach suitable for it. The next step of our work will be to experiment with more complex, possibly real cases of hypotheses based legal reasoning.

6. REFERENCES

- [1] A. Ciampolini, E. Lamma, P. Mello, and P. Torroni. Laila: a language for coordinating abductive reasoning among logic agents. *Computer Languages*, 27(4):137–161, 2001.
- [2] A. C. E. Lamma, P. Mello, F. Toni, and P. Torroni. Co-operation and competition in alias: a logic framework for agent that negotiate. *Annals of*

Mathematics and Artificial Intelligence, 37(1-2):65–91, January 2003.

- [3] G. Maggenti. Un linguaggio di coordinamento per agenti logici abduitivi. Master's thesis, Dept. of Computing, University of Pisa, 2004. (in Italian <http://etd.adm.unipi.it/theses/available/etd-03302005-174813/>).
- [4] C. S. Peirce. *Collected Papers*. Harvard University Press, Cambridge, 1931–1935.
- [5] H. Prakken and G. Sartor. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-Classical Logics*, 7(1), 1997.
- [6] H. Prakken and G. Sartor. The role of logic in computational models of legal argument: A critical survey. In A. Kakas and F. Sadri, editors, *Computational Logic: Logic Programming and Beyond. Essays In Honour of Robert A. Kowalski*, volume II of *LNCS 2048*, pages 342–380. Springer, 2002.
- [7] G. Sartor. *Intelligenza Artificiale e Diritto: un'Introduzione*. Giuffrè, Milano, 1995. In Italian.
- [8] M. Wooldridge. Intelligent agents. In G. Weiss, editor, *Multiagent Systems A Modern Approach to Distributed Artificial Intelligence*, pages 27–77. MIT Press, 2000.
- [9] M. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10(2):115–152, 1995.