e-Everything: e-Commerce, e-Government, e-Household, e-Democracy

14th Bled Electronic Commerce Conference

Bled, Slovenia, June 25 - 26, 2001

Temporality, Priorities and Delegation in an E-Commerce Environment

Luís Brito, Paulo Novais, José Neves

Universidade do Minho, Departamento de Informática
Campus de Gualtar, 4710-057 Braga, Portugal
LBrito@alfa.di.uminho.pt, Pjon@di.uminho.pt, JNeves@di.uminho.pt

Abstract

Logic presents itself as a major tool in the development of formal descriptions for agent-based systems; indeed, Logic Programming (LP) and specially Extended Logic Programming (ELP) provide a powerful tool for the development of such systems, besides being mathematically correct (and subject to proof) are easy to prototype.

On the other hand Electronic Commerce (EC) poses new challenges in the areas of Knowledge Representation and Reasoning (KRR) and formal modelling, where specific agent architectures are mandatory. In this work such a problem will be approached by the Experience-Based Mediator (EBM) agent one, particularly suited to take into account the argumentation processes that are inherent to any EC deal

The last but not least, although logic has been successfully used in the areas of argumentation (specially, legal argumentation), the reasoning process that happens before hand is rarely stated. In EC scenarios, such process takes into account features such as temporality, priorities, delegation, making use of incomplete information, and leading to feasible EC systems.

1. Introduction

The amount of ambiguity present in real-life negotiations is intolerable for automatic reasoning systems. Formalizing the process that opposes, during negotiation, intelligent agents is, therefore, extremely important. Logic, and especially, Extend Logic Programming (ELP) [2] poses itself as a powerful tool to achieve both the desired formality without compromising comprehension/ readability, and the ability to easily build an executable prototype. Logical formulas are extremely powerful, unambiguous and possess a set of interesting advantages [7]:

> *Expressing information in declarative sentences is far more modular than expressing it in segments of computer programs or in tables. Sentences can be true in a much wider context than specific programs can be used. The supplier of a fact does not have to understand much about how the receiver functions or how or whether the receiver will use it. The same fact can be used for many purposes, because the logical consequences of collections of facts can be available.*

However, in a dynamic environment such as the one found in Electronic Commerce (EC), the simple use of logical formulas is not enough. The use of non-monotonic characteristics is self-evident (which is in someway found in ELP). An agent believes that something is true at a given time but it may conclude differently at a later time [9].

In extended logic programs, queries may not be answered with a simple *true* or *false*. To represent incomplete knowledge about the world, the possibility for the *unknown answer* exists. To enable this to happen, extended logic programs contain two kinds of negations: ¬ (called *classical*, *strong* or *explicit negation*) and *not* (called *negation-by-failure*; i.e., the proof fails). In general logic programs, negative information is provided by the *closed-world assumption* (i.e., everything that can not be proven to be *true* is *false*), however, in extended logic programs, that is not so. In ELP a query may fail due to the fact that *information is not available to support it* or, on the other hand, it may fail due to the fact that *negation* succeeds. The Knowledge Base (KB), which serves as the basis for the agent's reasoning, can be seen has an extended logic program (Π) which is a collection of rules with the form:

$$L_0 \leftarrow L_1, ..., L_m, not\ L_{m+1}, ..., not\ L_n$$

where L_i $(0 \leq i \geq n)$ is a *literal* (i.e., formulas of the form p or $\neg p$, where p is an atom). This general form is reduced to $L_0 \leftarrow$ (also represented as L_0) in the case of facts.

The use of logic to formalize the reasoning behind the negotiation process, completes previous work in the same area. The reasoning strategies behind each agent (hereby addressed) follows a quantification of anthropomorphic features such as agreement and gratitude, during strategic planning [4], and gives way to a formal definition of the argumentative process.

The strategy to get a consistent and sound approach with the use of agents in EC is based on:

1. **Architecture**: to define and specify the agent's modules or functionalities, to design the flow of information (e.g., BDI [12], Experience-Based Mediator (EBM) agent [10]);

2. **Process quantification**: to quantify each metric and/or sub-process with which the agents may have to deal with. To establish the mechanisms and protocols for an objective approach to any kind of problem;
3. **Reasoning mechanism**: each agent is in need of a formal set of rules that will serve as the main guidelines for the negotiation processes. A set of parameters are to be taken into account/evaluated before the agents start any deal;
4. **Process formalization**: the process of argumentation needs to proceed via a formal specification to a consistent implementation in order to set the agents to act/react in a reasonable (logic) way. Arguing during a negotiation has many similarities to legal arguing [11] and logic presents itself, once again, as a powerful specification and implementation tool.

Taking the EBM agent as the selected architecture, the reasoning that lies under the *negotiation procedures* is the main object of formalization. Temporality introduces a temporal validity for the base clauses (facts) in the KB, which is important when taking into account the non-destructive (non-monotonic) principles of knowledge assimilation. Priorities are important when reasoning over the set of base clauses in order to justify/explain some action. Delegation is the underlying principle in the use of proxy agents.

The main contributions of this work are: (i) the definition of a common ground to situate the agent's reasoning mechanisms in EC environments; (ii) the use of formal tools (logic) to describe the rational behaviour of agents involved in EC; (iii) the description of a reasoning mechanism necessary for a consistent and sound development of agents for EC; and (iv) the use of incomplete information in the reasoning process.

2. Right to Deal - Basic Assumptions

The KB of an agent is made of a series of facts and rules. Facts provide the basic information, make the agent's *object knowledge level*, and feed the mechanisms at the meta level (i.e., at the *reasoning knowledge level*) in order to make an agent to behave properly.

During a negotiation process, each agent, although being able to deal with a counter-part, may be inhibit to do so. Therefore, a distinction must be established between capability (i.e., an agent has the necessary expertise to do something) and right (i.e., an agent has the capability to do something and it can proceed that course of action) [8].

In the case of an EBM agent, it is assumed that it has the ability to deal with every product, under any scenario. However, any agent has its behaviour conditioned by the right to deal premise. It will be considered the predicates *capability-to-deal: Product, Conditions, Counterpart* $\rightarrow$ *{true, false}* (representing the capability to deal), and *right-to-deal: Product, Conditions, Counterpart* $\rightarrow$ *{true, false}* (representing the right to deal), where *Product, Conditions* and *Counterpart* stand, respectively, for the product to be traded, the conditions associated to that operation and, the counter-part agent involved in the deal. It may now be stated that:

$\vdash \forall_{Product} \forall_{Conditions} \forall_{Counterpart}$ *capability-to-deal (Product, Conditions, Counterpart)*

where $\vdash$ and $\forall$ stand, respectively, for the "derivability relation" and the quantifier "for all".

The presence of such knowledge in the KB of an agent, can be taken as implicit. The knowledge about the capability to deal is replaced in the KB by the right to deal (*right-to- deal: Product, Conditions, Counterpart* $\rightarrow$ *{true, false}*).

3. Preliminaries

Before approaching each of the system's entities in order to formalize it using ELP, the introduction of a series of concepts is in order, namely the ones for theorem solver, restriction/invariant and incomplete information, that pose the basic building blocks for further definitions.

Notation (Implicit Agent)

Factual clauses represented by $ag_x : P$, and rule clauses represented by $ag_x : P \leftarrow Q$, present in the KB of agent ag_x, may be denoted as P and $P \leftarrow Q$, without loss of meaning, whenever the owner agent is implicit.

3.1 Invariants

A restriction or invariant is a condition or a set of conditions that are to be maintained by an agent on a permanent basis. In the general case, the theory of invariants is defined in terms of predicates given by the theory of rules, in terms of a meta theoretic definition.

Definition 1 (Invariants)

Invariants are represented at the agent's KB through clauses of the form *A:+restriction::P*, where *A*, *+restriction* and *P* stand, respectively, for a agent, the invariant's type and the invariant itself.

3.2 Null Values

Typically, EBM agents act in situations where dealing with a given agent is forbidden or, in some way, the set of conditions to be followed in a deal are not completely defined. These situations involve the use of *null values*. A special theorem solver was developed in order to cope with this kind of information.

Definition 2 (LP Theorem solver for Incomplete Information)

Taking factual clauses (represented by P) and rule clauses (represented by $P \leftarrow Q$) as the components of the KB of agent ag_x, the predicate $demo_{LPI} : T, V \rightarrow \{true, false\}$, where T, V and {*true*, *false*} stand, respectively, for a logical theorem, the valuation of theorem T and the set of possible valuations for $demo_{LPI}$. The *LP* theorem solver for incomplete information with null values over the KB of an agent, may therefore expressed by the following set of rules:

$$demo_{LPI}(T, true) \leftarrow T.$$
$$demo_{LPI}(T, false) \leftarrow \neg T.$$
$$demo_{LPI}(T, unknown) \leftarrow not\ T, not\ \neg T.$$

With the use of incomplete information with null values, a simple 3-valued logic is set into place. Using this framework, it is now possible to assert the conditions under which a given product or service may be traded.

The situation where the ability to trade product *P* with agent *Y* is known, but the trading conditions are not defined, may be overcome with the use of a *null value from an unknown set of values* [2].

Definition 3 (Unknown Conditions taken from an Unknown Set of Values)

An "unknown conditions" situation arises when an agent can not express in its KB the right to deal with a given counter-part, using a set of conditions taken from an unknown set of possible values). Taking predicates $null_{unknown\text{-}set}: N \rightarrow \{true, false\}$ (expressing the null value statement), $right\text{-}to\text{-}deal: P, C, CP \rightarrow \{true, false\}$ (expressing the right to deal) and $exception_{rtd}: P, C, CP \rightarrow \{true, false\}$ (expressing the situation to the right to deal), where *N*, *P*, *C*, *CP* and {*true, false*} stand, respectively, for the null value definition, the product to be traded, the conditions to consider during negotiation, the counterpart agent, and the valuation set for each predicate, the KB of an agent has to be augmented by the rules:

$$exception_{rtd}(P, -, CP) \leftarrow null_{unknown\text{-}set}(X), right\text{-}to\text{-}deal(P, X, CP).$$
$$\neg exception_{rtd}(P, C, CP) \leftarrow not\ right\text{-}to\text{-}deal(P, C, CP), not\ exception_{rtd}(P, C, CP)$$

where predicate $\neg right\text{-}to\text{-}deal: P, C, CP \rightarrow \{true, false\}$ stands for the negative information on the right to trade *P*, under the market conditions *C* and counterpart *CP*. The KB of an agent must contain an instantiation of $null_{unknown\text{-}set}$ (e.g., $null_{unknown\text{-}set}$ (*cond*)) and *right-to-deal()* clauses which may use the null value (e.g., *right-to-deal(p4,cond,cp2)*).

4. The Logic Structure behind Negotiation

As it was previously stated, the set of the most important features that intervene in the negotiation process, will now be object of study. These include temporality, priorities and delegation. Through assessment, an agent may weight its knowledge base, its temporal validity and relative priorities, and then deciding if delegation is in order.

The general process of negotiation must be clearly distinguished from the argumentative one. The process of argumentation is tightly coupled with the process of logically founded attack on the arguments put forward by a counterpart. It deals with price-formation issues and deal finalization. On the other hand, negotiation is a wider concept that is coupled with specific forms of reasoning, dealing with the high-order, pre-arguing relationships, that may be established among agents.

In order to establish a common ground for the formalization of each concept, a logical theory (on which the KB of each agent is based upon) is to be defined, that will serve as a the backbone of a broader one for negotiating proposes.

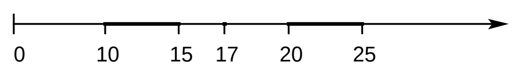
Definition 4 (A Logical Theory for Negotiating Agents)

A Logical Theory for Negotiating Agents is defined as the quadruplet $TNA = \langle R, C, BP, \pi \rangle$ where *R, C, BP* and π stand, respectively, for the set of predicates on the right to deal (*right-to-deal: Product, Conditions, Counterpart* $\rightarrow$ *{true, false}*), the set of invariants (*A:+restriction::P*), the set of behavioural predicates (including the theorem proffers) and a non-circular order relation that states that if $P \pi Q$, then *P* occurs prior *Q*; i.e., having precedence over *Q*.

4. 1 Temporality

The concept of temporality is linked to the temporal validity of possible inferences over the KB of an agent; i.e., a fact may be valid only on a well-defined time period. Taking a non-destructive KB and a non-monotonous logic, different conclusions may be reached when the temporal validity of information is taken into account (e.g. John has the right to deal with Paul but only form 10/05/2001 to 12/05/2001).

Taking the set *R* (composed of *right-to-deal* clauses) from the logical theory *TNA*, an extension is to be made in order for these elements to encompass *temporal* validity. Therefore, an agent will reason about validity taking into account the information present at the fact level. An example of validity, for a specific clause, is shown in Figure 1.



right_to_deal (p1,[c1,c2],john)::[[10,15},[17,17],[20,25]].

Figure 1: *Example of time validity for a right-to-deal clause*

Definition 5 (Clauses with Temporality)

A factual clause, represented as *P.*, where *P* is an atomic formula, is represented, to encompass temporal validity, as $P::[i_1,i_2,...,i_n].$, where $i_j=[t_a, t_b]$ is one of the following elements:

1. temporal instant

$t_a = t_b$, $t_a, t_b \geq 0$ with $t_a, t_b \in TD$.

$TD = \{t | t \in N_o\} \cup \{forever\}$, where *forever* represents the *end of times*.

2. temporal interval

   $t_a < t_b$, $t_a \geq 0$ with $t_a, t_b \in TD$.

   $TD = \{t | t \in N_o\} \cup \{forever\}$, where *forever* represents the *end of times*.

   In the case where positive and negative information is present in the KB, set *R* of theory *TNA* should be consistent; i.e., the following condition should be verified:

$$\exists P :: T_1 \wedge \exists P :: T_2 \rightarrow T_1 \cap T_2 = 0$$

Having restructured the factual clauses in the KB of each agent, a demonstrator/theorem solver has to be constructed in order for the agent to deal with the new time semantics. This theorem solver is taken as an evolution of the one presented for dealing with *incomplete information*.

Definition 6 (A LP Theorem Solver for Incomplete and Temporal Information)

Taking factual clauses with temporal validity (represented by $P::[i_1, i_2, \ldots, i_n]$.) and rule clauses (represented by $P \leftarrow Q$. and being read as "*P if Q*") as the components of the KB present in each agent, the predicates $demo_{LPIT}: T, CT, V \rightarrow \{true, false\}$, where *T, CT, V* and *{true, false}* stands, respectively, for a logical theorem, the current time, the theorem valuation (true, false, unknown) and the possible valuations for the $demo_{LPIT}$ predicate, represents the LP theorem solver for incomplete and temporal information over the KB, governed by the following set of rules:

$demo_{LPIT}(P, CT, true) \leftarrow P::T,$
$\quad in_{time}(CT, T).$
$demo_{LPIT}(P, CT, false) \leftarrow P::T,$
$\quad \neg in_{time}(CT, T).$
$demo_{LPIT}(P, CT, false) \leftarrow \neg P::T,$
$\quad in_{time}(CT, T).$
$demo_{LPIT}(P, -, unknown) \leftarrow not\ (P::-),$
$\quad not\ (\neg(P::-)).$
$in_{time}(Ct, [T]) \leftarrow in_{interval}(CT, T).$
$in_{time}(Ct, [T_1|T]) \leftarrow in_{interval}(CT, T_1),$
$\quad in_{time}(CT, T).$
$in_{interval}(CT, [T_a, T_b]) \leftarrow T_a \geq T_b,$
$\quad T_a \geq 0,$
$\quad CT \geq T_a,$
$\quad CT \leq T_b.$
$in_{interval}(CT, [T_a, forever]) \leftarrow T_a \geq 0,$
$\quad CT \geq T_a.$

where predicates in_{time}*:CT, LT* $\rightarrow$*{true, false}* and $in_{interval}$*: CT, TI* $\rightarrow$ *{true, false}* stand, respectively for the verification of presence of time *CT* in the list of validity intervals *LT* and for the verification of presence of time *CT* in time interval (or instant) *TI*.

4. 2 Priorities

As previously stated, logical theory TNA introduces a non-circular ordering (trough relation π) on the clauses present in the KB of an agent. However, this ordering, based on the temporal sequence of KB assertion, is insufficient for expressing some inferences. As it is known, real-world agents (humans) are able to prioritise the treatment of knowledge in their reasoning (e.g. the reasoning that takes place in a negotiation, especially over the *right to deal*). This need for extended priorities may also be seen in the posterior arguing stage.

Suppose the following set of factual clauses was present at the KB of an agent:

right-to-deal(p1, somecondition ,john).
right-to-deal(p1 ,[c1] ,charles).
right-to-deal(p1, [c1,c2], paul).

Having in mind that the KB of an agent is conditioned to the ordered, non-circular, logical theory *TNA*, this set of facts can be interpreted, due to the relative ordering of clauses, as: there is the right to deal product *p1*, with some undefined condition with *john*, followed by the right to deal product *p1*, with condition *c1* with *charles* and, following the two previous right statements, there is the right to deal *p1* with condition *c1* and *c2* with *paul*. Any change is the priority treatment of clauses (e.g. the clauses related to *paul* have an higher priority than those that relate to *john*), would lead to a restructuring of the KB, in order for the clause ordering established by relation π of *TNA* to express the new priority semantics. As expected, this course of action is unfeasible in systems where large sets of clauses are present.

The solution, for a feasible priority treatment, lies in the embedding of priority rules on the KB of each agent. Therefore, logical theory *TNA* is to be changed into a new logical theory *(TNAP)* in which the organization of factual clauses is given by the semantics of *priority rules*.

Definition 7 (A Logical Theory for Negotiating Agents with Priorities)

The Logical Theory for Negotiating Agents with Priorities is defined as *TNAP=*$\langle$ *R,C,BP,PR,*π $\rangle$ where, *R*, *C*, *BP*, *PR* and π stand, respectively, for the set of predicates on the right to deal (*right-to-deal:Product,Conditions,Counterpart* $\rightarrow$ *{true, false, unknown}*), the set of assertion restrictions/invariants (*A:+restriction::P.*), the set of behavioural predicates (including all demonstrators/theorem solvers), the set of embedded priority rules and the non-circular order relation established among the different clauses in a KB that derives from the time of their insertion. Relation π *de*termines, in the case of *P* π *Q*, that *P* is earlier than *Q*, thus ordering the set of clauses, providing for a fail-safe priority mechanism under the one provided by the set *PR*.

Although priorities can be established between single clauses, it is usual, at least as a first-level approach, to consider priorities among *bodies of knowledge* (e.g. information about *mary* as priority over information about *john*). These *bodies of knowledge* are nothing more than a high-level classification of factual clauses. Notice, however, that this classification has variable *granularity*, giving way to a per-clause priority if so needed (with the consequent increase in complexity).

Therefore, the factual clauses dealing with the right to deal, already expanded to include temporal information, are now to be once again expanded to include a *body of knowledge* classification.

Definition 8 (Clauses with Temporality and Body of Knowledge Classification)

Taking $P::[i_1,i_2,...,i_n]$. as a clause, where P is the simple factual clause and $[i_1,i_2,...,i_n]$ is the temporal validity for the clause, then $BK::P::[i_1,i_2,...,i_n]$ represents a clause with temporality and body of knowledge classification, where BK stands for the body of knowledge to which P is associated.

Having defined the concept of *body of knowledge* and the structure of the factual clauses that express the right to deal, the structure of the priority rules present at set *PR* in logical theory *TNAP* may now be stated.

Definition 9 (Priority Clauses)

Taking BK_i and BK_j ($i \neq j$) as two bodies of knowledge, and supposing that the knowledge associated with BK_i has priority over the knowledge associated with BK_j, then the predicate to insert into set *PR* of logical theory *TNAP* expressing such semantics is $priority: BK_a, BK_b \rightarrow \{true, false\}$ (where BK_a and BK_b two bodies of knowledge) and the particular instantiation is $priority(BK_i,BK_j)$.

Set *PR* should be consistent; i.e., it should verify the following condition:

$$\forall BK_i, BK_j \; priority(BK_i,BK_j) \rightarrow \neg successor(BK_j,BK_i)$$

$$successor(X,Y) \leftarrow priority(X,Y).$$

$$successor(X,Y) \leftarrow priority(X,Z), successor(Z,Y).$$

In order for the priority information to be used in the reasoning of agents, the demonstrator/theorem solver that is able to deal with incomplete and temporal information (see Definition 6) has to be expanded.

Definition 10 (A LP Theorem Solver for Incomplete and Temporal Information with Priorities)

Taking factual clauses with temporal validity and body of knowledge classification (represented by $BK::P::[i_1,i_2,...,i_n]$.) and rule clauses (represented by $P \leftarrow Q$. and being read as "*P if Q*") as the components of the KB present in each agent, the predicate $demo_{LPITP}:T,CT,V \rightarrow \{true, false\}$, where T, CT, V and $\{true, false\}$ stands, respectively, for a logical theorem, the current time, the theorem valuation (true, false or unknown) and the possible valuations for the $demo_{LPITP}$ predicate, represents the LP theorem solver for incomplete and temporal information over the KB, governed by the following set of rules:

$$demo_{LPITP}(P, CT, true) \leftarrow priority(BK_1, BK_2),$$

$test_{priority}(BK_1, BK_2, P,T),$
$in_{time}(CT, T).$
$demo_{LPITP}(P, CT, false) \leftarrow \quad priority(BK_1, BK_2),$
$test_{priority}(BK_1, BK_2, P,T),$
$\neg in_{time}(CT, T).$
$demo_{LPITP}(P, CT, false) \leftarrow \quad priority(BK_1, BK_2),$
$ntest_{priority}(BK_1, BK_2, P,T),$
$in_{time}(CT, T).$
$demo_{LPITP}(P, -, unknown) \leftarrow \quad priority(BK_1, BK_2),$
$not\ test_{priority}(BK_1, BK_2, P,-),$
$not\ \ ntest_{priority}(BK_1, BK_2, P,-).$

$test_{priority}(BK_1, -, P,T) \leftarrow BK_1::P::T).$
$test_{priority}(-, BK_2,, P,T) \leftarrow BK_2::P::T).$

$ntest_{priority}(BK_1, -, P,T) \leftarrow \neg (BK_1::P::T).$
$ntest_{priority}(-, BK_2, P,T) \leftarrow \neg (BK_2::P::T).$

where predicates in_{time}: $CT, LT \rightarrow \{true, false\}$, $test_{priority}$: $BK_a, BK_b, P, T \rightarrow \{true, false\}$ and $ntest_{priority}$: $BK_a, BK_b, P, T \rightarrow \{true, false\}$ stand, respectively for the verification of presence of time CT in the list of validity intervals LT, the prioritised demonstration of theorem P for the bodies of knowledge BK_a and BK_b and the prioritised demonstration of theorem P through negative information for the bodies of knowledge BK_a and BK_b.

4.3 Delegation

Delegation can be seen as the delivery (assimilation) of a valid negotiation from one agent to another. The knowledge concerning the right to deal must be taken into account. Negotiation tasks may only be delivered to a third party if there is sufficient knowledge relating to the *right to deal* with that same agent.

Delegation acts as a way to undertake indirect negotiations; i.e., use a proxy agent taking advantage of its particular characteristics, such as gratitude debts and agreements established among the proxy and other agents. Therefore, formalizing the delegation process is equivalent to formalizing the generation of a "middle-man" approach to business.

Definition 11 (Delegation)

Taking logical theory *TNAP* over which the KB of an agent is built and the theorem solver for incomplete and temporal information with priorities ($demo_{LPITP}$*:P,CT,V→{true, false}*), agent ag_x can delegate a negotiation activity on an agent *Ag*, in order for this last one to act as a proxy element, as long as its reasoning is regulated by the logical structure of predicate *delegate: What, Conditions, CounterPart, Ag, CurrentTime →{true, false, unknown}*:

ag_x : delegate(P,C,CP,Y,CT)←

ag_x : $demo_{LPITP}$(right-to-deal(P,C,CP), CT, true),

ag_x : $demo_{LPITP}$(right-to-deal(P,-,Y), CT, true),

Y: $valid_{assimilation}$(Y: right-to-deal(P,C,CP), CT).

Then, $\forall ag_y$*:* $ag_y \in$ *Agents*, where *Agents* stands for the set of valid agents, it follows:

ag_y: $valid_{assimalation}$(X, CT) ← ag_y : $demo_{LPITP}$(X, CT, true).

ag_y: $valid_{assimalation}$(X, CT) ← ag_y : assert(X::[[CT, CT]]),

ag_y : solutions(Z, ag_y : +restriction::Z, S),

ag_y : $demo_{restrictions}$(S, CT, true).

ag_y: $valid_{assimalation}$(X, CT) ← ag_y : retract(X::[[CT, CT]]).

ag_y : $demo_{restrictions}$([], -, true).

ag_y : $demo_{restrictions}$([R_1],CT, V)← ag_y : $demo_{LPITP}$(R_1,CT, V).

ag_y : $demo_{restrictions}$([R_1|T],CT, true)← ag_y : $demo_{LPITP}$(R_1,CT, true),

ag_y : $demo_{restrictions}$(T, CT, true).

ag_y : $demo_{restrictions}$([R_1|T],CT, false)← ag_y : $demo_{LPITP}$(R_1,CT, V_1),

ag_y : $demo_{restrictions}$(T, CT, V_2),

$V_1 \neq V_2$.

where predicates $valid_{assimilation}$*: P, CT→{true, false]*, *solutions: A, C, S→{true, false}* and $demo_{restrictions}$*: S, CT,V→{true, false}* stand, respectively, for: the validity of assertion of *X* in the agent's KB at time *CT*; the collector of elements into *S* with archetype *A* and complying with conditions *C*; and the verification of compliance of the set of elements *S* (valuation of *S* -- *V*), in time instant *CT*. The *assert: P→{true, false}* predicate is responsible for the classification of *P* in terms of body of knowledge.

5. Reasoning Examples

Taking the previously defined formalizations, the presentation of a small set of examples, representing the main reasoning situations in which an agent is involved, is in order.

Assume the following KB, defined according to the non-circular theory *TNAP*:

$null_{unknown\text{-}set}(somecondition).$

$bk_1 : right\text{-}to\text{-}deal(p1, somecondition, cp1) :: [[0, 5], [8, forever]].$

$bk_1 : \neg\, right\text{-}to\text{-}deal(p1, somecondition, cp1) :: [[6, 7]].$

$bk_2 : right\text{-}to\text{-}deal(p1, [c1], cp2) :: [[0, forever]].$

$bk_2 : right\text{-}to\text{-}deal(p3, [c1, c2], cp2) :: [[0, 10], [30, 40]].$

$bk_3 : right\text{-}to\text{-}deal(p5, [c1], cp3) :: [[0, forever]].$

$bk_3 : right\text{-}to\text{-}deal(p6, [c1, c3], cp3) :: [[0, forever]].$

$bk_1 : \neg\, right\text{-}to\text{-}deal(P, C, CP) :: T \leftarrow not\ bk_1 :: right\text{-}to\text{-}deal(P, C, CP) :: T,$

$\qquad not\ bk_1 :: exceptionn_{rtd}(P, C, CP) :: T.$

%exceptions

$bk_1 :: exception_{rtd}(P, -, CP) :: T \leftarrow null_{unknown\text{-}set}(X),$

$\qquad bk_1 :: right\text{-}to\text{-}deal(P, X, CP) :: T.$

%priorities

$priority(bk_2, bk_1).$

$priority(bk_1, bk_3).$

Product p1 can be negotiated with counter-part agent cp1, however, due to the null from an unknown set, the set of conditions is not defined (e.g., it can not be stated that this product can be negotiated with conditions [c1, c3], because this fact is unknown).

In terms of time validity, the negotiation of p1 with counter-part agent cp1 is divided into three time intervals. Two of these intervals define positive information (the right to deal exists) and one of them defines negative information (the right to deal does not exist). In the case of product p3 being negotiated with conditions [c1, c2] and counter-part cp2, two time intervals are defined where the right to deal exists. Outside these intervals, the right to deal is unknown.

Three *bodies of knowledge* have been defined taking into account the counter-part agent. Although the facts of agent cp1 appear before those of cp2 and those of an undefined *someone* agent, the priority clauses must be taken into account. The priority semantics is defined to infer over the clauses of cp2, then over those of someone and, only then, over those of cp1.

The KB can be queried through the theorem solvers. For instances,

$?demo_{LPITP}(right\text{-}to\text{-}deal(p1, [c1], cp2), 10, V).$ $\qquad V=true$

$?demo_{LPITP}(right\text{-}to\text{-}deal(p1, [c1, c2], cp1), 10, V).$ $\qquad V=unknown$

?$demo_{LPITP}$(right-to-deal(p1,somecondition,cp1),7,V). V=false

?$demo_{LPITP}$(right-to-deal(P,C,CP),10,true). P={p1, p3, p1, p5, p6}
C={[c1], [c1, c2],
somecondition, [c1], [c1, c3]}
CP={cp2, cp2, cp1,cp3,cp3}

The second column expresses the variable instantiations for which the query succeeds. In the first and third queries, the valuation is reached through the simple use of, respectively, positive and negative information in the KB. The second query in valued as unknown due to the use of null values in the set of conditions; i.e., p1 can be negotiated with cp1 but the set of conditions can not be set to [c1, c2], although it is a possible value. The last query indicates the possible variable instantiations of product, conditions and counter-parts, taking into account that the right to deal must be stated as positive in the KB at instant 10.

Assume now a new set of clauses representing the KB of agent ag_x and that of agent ag_y. Reasoning about delegation will now involve the set of restrictions embedded into the KBs. The clauses are:

ag_x : bk_1 : right-to-deal(p1,[c1], cp2):: [[0, forever]].

ag_x : bk_2 : right-to-deal(p2,[c3, c4], cp3):: [[0, 50]].

% exceptions ag_x

%theorem proffers ag_x

% priorities

ag_x: priorities(bk_1, bk_2).

ag_y : bk_1 : right-to-deal(p2,[c3, c4], cp3):: [[0, 60]].

ag_y : bk_1 : right-to-deal(p2,[c5], cp4):: [[0, 10]].

ag_y :money(900).

% exceptions ag_y

%theorem proffers ag_y

% priorities

ag_y: priority(bk_1, bk_2).

Agent ag_x is able to negotiate product p1, taking conditions [c1] with counter-part agent ag_y permanently. In the case of product p2, conditions [c3, c4] are established for counter-part agent cp3, but only for interval [0,50]. The knowledge about the right to deal with ag_y overpowers the knowledge about cp3.

Agent ag_y is able to negotiate product p2, taking conditions [c3, c4] with counter-part agent cp3, but only on interval [0,60]. Furthermore, it is able to negotiate product p2, taking conditions [c5] with counter-part agent cp4, but only on interval [0,10]. Agent ag_y has 900 monetary units expressed in its KB and a new assertion is conditioned to the existence of 1000 monetary units (due to an assertion restriction). Priority rules establish that the knowledge about cp3 overpowers that of cp4.

The KB can be queried in order to determine the validity of a delegation process:

? ag_x: delegate(p1,[c1],cp2, ag_y,10).	*false*
? ag_x: delegate(P,C,cp2, ag_y, 10).	*P={p1}*
	C={[c1]}

The second column expresses possible variable valuations or the valuation for the query itself. In the first query, although the right to deal is well established in agent ag_x, it is impossible to assert the necessary knowledge in the proxy agent (ag_y) due to the assertion restriction. In the second query, the delegation on agent ag_y for a negotiation with cp2, at time instant 10, is only possible for product p1 and conditions [c1].

6. Related Work

The establishment of multiagent frameworks for EC falls short of a formal justification. The definition of generic agent architectures, such as the BDI and sub-sumption ones, reveals itself as inappropriate (in their raw form) to deal with the specificities of a field such as EC or Electronic Business (EB). Therefore, the definition of an architecture that combines general characteristics with specific modules is in order, giving way to the EBM agent architecture [10].

The work in formalization of reasoning and agent interaction through logic is not new. However, the use of such methodology to the specficic area of EC reveals itself as being of paramount importance.

Process formalization through logic has been directed, through the work of *Prakken* and *Sartor* ([11], [13], [14]) and *Kowalski* and *Toni* ([5], [6]) to the field of argumentation. In particular, the field of legal argumentation.

The concept of rights is central to the reasoning process of an EC-directed agent. Rights in multiagent systems have been approached by the work of *Norman*, *Sierra* and *Jennings* [8] through a logical framework.

Negotiations in economic fields must take into account incomplete information. Work has been done by *Analide* and *Neves* [1] in the process of formalization of incomplete information through logic. However, it was applied to the EC field by work done by *Brito, Novais and Neves* [3].

7. Conclusions

Logic reveals itself as a simple and powerful tool for process formalization. The use of LP and, furthermore, the use of ELP provides a mathematical background to otherwise subjective statements. Such tools provide the means for mathematical proof of statements, over the KB of each agent, and the possibility to create a working prototype.

Taking into account the advantages of ELP, an agent architecture must be devised for EC. As previously stated, generic architectures fall short of mechanisms to deal with the specificities of a field such as that of EC. Therefore, the EBM agent architecture was used, containing modules that are specially suited to deal with particularities such as past-experiences, temporality, priorities and delegation.

The set of formal tools here presented serve as the basis for each agent to decide upon the process of starting (or not) a negotiation process with a specific counterpart. Each agent reasons over its set of knowledge clauses coming (or not) to the decision of engaging another agent. Therefore, this phase involves the reasoning of each agent about its relationship with others. The real "conflict" and argument evaluation by each part takes place only on the next step of the 4-step strategy here proposed for the development of EC-directed agents.

For an EC-based agent, logic reasoning is important due to the necessity of the establishment of a feasible and justifiable negotiation process. However, simple logic does not convey some of the important characteristics of a real-world business. The inability to deal with some other agent, the knowledge about dealing with an agent but without knowing the conditions of a specific deal, or even, dealing with an agent without knowing the conditions but having them restricted to a specific set, are common situations. This can be easily approached by the use of null values. Furthermore, an EC-based agent must be able to deal with knowledge that is only valid on specific states (i.e., the knowledge is only valid in specific time periods).

Formalization of the reasoning process developed by an EBM agent must be seen as a gradual process. First, the basic elements of reasoning must be established (the right to deal) and the inference mechanisms over them are to be defined. These inference mechanisms must evolve from simple LP reasoning to ELP reasoning with incomplete information and, from there, to ELP reasoning with temporal and priority considerations.

In order to introduce priorities, the use of a body of knowledge for classification provides for the desired granularity over the KB. However, priority rules that are embedded into the KB, such as the ones found in the present work, must be constrained in order to ensure KB soundness.

Delegation introduces non-linearities in the reasoning process in order to condition the strategic planning of an agent. Once again, logic serves as a powerful tool to model subjective elements.

Acknowledgements

The work of Luís Brito was supported by a PRAXIS grant (PRAXIS XXI/BD/21541/99). The work of Paulo Novais was supported by the PRODEP's project (5.2/N/189.017/98). The work of José Neves was supported by a PRAXIS grant (PRAXIS Project/P/EEI/13096/98).

References

[1] Analide, C., Neves, J., Representation of Incomplete Information, in CAPSI, Conferência da Associação Portuguesa de Sistemas de Informação, Universidade do Minho, Guimarães, Portugal, 2000.

[2] Baral, C., Gelfond, M., Logic Programming and Knowledge Representation, Research Report, Computer Science Department, University of Texas at El Paso, El Paso, Texas, 1994.

[3] Brito, L., Novais, P., Neves, J., A General Agent-Based Architecture for Production Planning in Electronic Commerce Scenarios, ESM2000 - European Simulation Multiconference, Ghent, Belgium, 2000.

[4] Brito, L., Novais, P., Neves, J., Mediation, Agreement and Gratitude in Strategic Planning for Virtual Organisations, ICEIS2000 - 2nd International Conference on Enterprise Information Systems, Stafford, UK, 2000.

[5] Kowalski, R., Toni, F., Abstract Argumentation, 1996.

[6] Kowalski, R., Toni, F., Argument and Reconciliation, 1996.

[7] McCarthy, J., Programs with Common Sense, Proceedings of the Teddington Conference on the Mechanization of Thought Processes, 75-91, London, 1959.

[8] Norman, T., Sierra, C., Jennings, N., Rights and Commitment in Multi-Agent Agreements, Proceedings 3rd International Conference on Multi-Agent Systems (ICMAS-98), Paris, France, 222-229, 1998.

[9] Neves, J. A Logic Interpreter to handle time and negation in logic data bases. In Proceedings of the ACM 1984 Annual Conference, pages 50-54, San Francisco, California, USA, 1984.

[10] Novais, P., Brito, L., Neves, J., Experience-Based Mediator Agents as the Basis of an Electronic Commerce System, Workshop 2000 - Agent-Based Simulation, Passau, Germany, 2000.

[11] Prakken, H., Logical Tools for Modelling Legal Argument, Doctoral Dissertation, Free University, Amsterdam, 1993.

[12] Russel, S., Norvig, P., Artificial Intelligence - A Modern Approach, Prentice-Hall, Inc., New Jersey, USA, 1995.

[13] Sartor, G., A Simple Computational Model for Nonmonotonic and Adversarial Legal Reasoning, Proceedings of the Fourth International Conference on Artificial Intelligence and Law, ACM Press, 192-201, 1993.

[14] Sartor, G., A Formal Model of Legal Argumentation, Ratio Juris 7, 212-226, 1994.