University of Thessaly

E-contracting Agents Reasoning with Assumptions

Georgios K. Giannikis and Aspassia Daskalopulu

Department of Computer and Communications Engineering,
University of Thessaly,
38221 Volos, Greece
{ggiannik,aspassia}@inf.uth.gr

Abstract. The work in this report is motivated from the need for assumption-based reasoning in normative systems, where realistically agents will have incomplete knowledge about their environment, and about other agents. The question we seek to address is whether it is possible for agents to identify appropriate assumptions dynamically in order to fill in informational gaps. We discuss and illustrate our proposals with reference to an e-commerce example. In our previous work, we argued that e-contracts could be represented as default theories and proposed a theoretical way in which such theories could be constructed automatically from initial Event Calculus representations. That proposal relied on determining what information could be proved from the agent's knowledge base, in order to decide whether it would serve as an assumption or not. In this report we present an incremental technique that can be used for this construction that enables the dynamic and ad hoc identification of candidate assumptions without resorting to proof. This idea is suitable for a computational implementation, and thus we have developed and discuss a prototype implementation. Finally, we survey other approaches to assumption-based or hypothetical reasoning. We broadly distinguish and discuss other approaches in those that employ assumptions statically and those that employ assumptions dynamically.

Keywords., Assumption-based Reasoning, Dynamic Assumptions, Default Logic, Normative Systems

May 2007

Table of Contents

1 Introduction

This report presents work conducted within a broader project that is concerned with the development of an open computational environment for electronic contracting. One of the issues of interest is to support temporal and defeasible reasoning, and to this end we proposed the representation of contracts as default theories, constructed dynamically from an initial Event Calculus (EC) representation [13, 14]. In our previous work, we also argued that, besides temporal and defeasible reasoning, such a representation of contracts enables agents to perform normative conflict detection and resolution.

Agents that use default theories (DfT) essentially reason with incomplete knowledge, by employing hypotheses. Hence, for an agent that employs assumptions, the following questions, arise naturally:

(i) What assumptions are applicable to fill in information gaps, i.e., what can be assumed by an agent to be true or false at various time points?

(ii) How do assumptions employed at some time point affect subsequent inferences? and

(iii) What happens when information that becomes available at some time point confirms or disproves assumptions made at previous times, i.e., how does new information affect previously drawn conclusions?

Here, we are concerned with the first of these issues, since the other two are addressed by employing the inference procedure of Default Logic (DfL).

Specifically, we present a technique that enables agents to identify appropriate candidate assumptions dynamically. We believe that such dynamic hypothetical reasoning can be useful in order for an agent to plan its activities in two modes: First, an agent may not (indeed cannot!) know the future, yet it may need to plan its activities on the basis of hypotheses that concern the future, i.e., on the assumption that certain events/actions will occur, or that certain causal relations will be effected, or that its partners will bear certain legal relations (obligations, permissions, powers, prohibitions). Second, an agent may not know everything about the past and present, i.e., the history so far, but yet needs to plan its activities by making hypotheses that concern the past/present.

We discuss and illustrate our proposal with reference to an e-commerce example, although it is, of course, more generally applicable. For the purposes of generality, although, as noted earlier, we have been constructing our e-contract representations as default theories from initial representations in Event Calculus [21], in what follows here, we do not employ Event Calculus explicitly.

The following section presents briefly our previous work on assumption-based reasoning and discusses its limitation with respect to a computational implementation. In section 3 we present an alternative way to reason hypothetically that is appropriate for the implementation of a computational tool. Section 4 illustrates this technique with an example and discusses a prototype implementation. In section 5 we discuss other approaches on assumption-based reasoning. Finally, section 6 summarizes our conclusions and directions for future research.

2 Preliminaries

The EC representation of an e-contract can be characterized as a triple (H, R, A), where H is a (possibly empty/incomplete) set of definitions for predicates H_L={Happens, HoldsAt, ¬HoldsAt} that denote domain-dependent historical information, R is a (possibly empty/incomplete) set of definitions for predicates R_L={Initiates, Terminates} that denote domain-dependent causal relations, and A is the (non-empty) set of definitions for the domain–independent predicates of EC, A_L={HoldsAt, ¬HoldsAt, Clipped, Declipped}, that is, $A=\{Y \leftarrow X1 \wedge \ldots \wedge Xk \mid Y \in A_L \text{ and } Xi \in A_L \cup H_L \cup R_L \cup T_L\}$[1].

For the sake of simplicity and without loss of generality, in this report we need not see in detail the EC representation and can instead think of the initial e-contract representation as comprising sentences of the form

$$Y \leftarrow X1 \wedge X2 \wedge \ldots \wedge Xk \qquad (1)$$

where Y and Xi ($1 \leq i \leq k$) are positive or negative literals (any variables are assumed universally quantified). From this representation we may construct dynamically a new one, in Default Logic [32].

A default rule (henceforth default) has the form P:J1,J2,...Jn/C, where P is the prerequisite, J={J1,J2,... Jn} is a set of justifications, and C is the derived consequent. The semantics of this rule is: If P holds and the assumption J is consistent with the current knowledge, then C may be inferred. Defaults of the form P:C/C are called normal. A DfT is a pair of the form (W, D), where W is a set of propositional or predicate logic formulae that represent currently available knowledge, and D is a set of defaults. A default is applicable to a deductively closed set of formulae $E \supseteq W$, iff $P \in E$ and $\neg J1 \notin E, \ldots, \neg Jn \notin E$. The set E is the extension of the DfT. We consider closed default theories, and derive extensions in the manner presented in [3], i.e., by maintaining syntactically consistent sets of formulae. An agent that derives conclusions on the basis of assumptions, by applying defaults, constructs the extension of its DfT incrementally. At each step i of the reasoning process, i.e. after the application of each default P:J1,...,Jn/C, the extension computed is a set of ground sentences $In(i)=In(i-1) \cup \{C\}$, and the set of assumptions employed, which should not turn out to be true, is $Out(i) = Out(i-1) \cup \{\neg J1, \ldots, \neg Jn\}$. For the first step of the process, i.e. for i=1, In(0)=W and $Out(0)=\emptyset$.

As discussed in our previous work [13], during the construction of a DfT, a sentence of the form (1) may be mapped to *any one* of the following defaults:

$X1 \wedge X2 \wedge \ldots \wedge Xk$: true / Y (that is, a justification-free default rule)

$X1 \wedge X2 \wedge \ldots \wedge Xk$: Y / Y (that is, a normal default rule)

$X1 \wedge X2 \wedge \ldots \wedge Xk\text{-}1$: Xk / Y

$X1 \wedge X2 \wedge \ldots \wedge Xk$: Xk-1 / Y

[1] Note that TL contains the first-order-logic predicates used to express temporal relations, i.e., $T_L=\{<, =, >, \geq, \leq\}$.

$X2 \wedge \ldots \wedge Xk : X1 / Y$

$X1 \wedge X2 \wedge \ldots \wedge Xk\text{-}2 : Xk\text{-}1, Xk / Y$

$X1 \wedge X2 \wedge \ldots \wedge Xk\text{-}1 : Xk\text{-}2, Xk / Y$

...

$X2 \wedge \ldots \wedge Xk\text{-}1 : X1, Xk / Y$

...

$true : X1, X2, \ldots, Xk\text{-}2, Xk\text{-}1, Xk / Y$ (that is, a prerequisite-free default rule)

That is, each sentence in the initial contract representation, which involves k conditions, corresponds to one of 2^k+1 defaults. The question that arises for the agent constructing the DfT is, which one of these 2^k+1 defaults should be chosen and employed in the inference procedure. This is tantamount to seeking to establish what assumptions are appropriate in order to fill in information gaps.

We presented an answer to this question in our previous [14], by providing a formal characterization of the DfT construction, relative to the currently available knowledge $H \cup R$. The W part of the DfT, is a copy of H, the possibly empty of incomplete historical information of the initial contract representation, i.e., it contains all currently available information about what holds and what happened. The set of defaults D of the DfT is constructed from the R and A parts of the initial contract representation, which are sentences of the form (1), as follows: The conclusion of each such sentence is mapped to the consequent part of each default, while its conditions may be mapped to the prerequisite or the justification part of each default, depending on what information is defined in the initial knowledge base: conditions that can be derived from $H \cup R$ are mapped to the prerequisite, while conditions that cannot be derived from K are mapped to the justifications.

Formally, an e-contract is the pair (W, D), where $W=H$ and D contains, for each definition $(Y \leftarrow X1 \wedge \ldots \wedge Xk) \in A \cup R$, (possibly semi-grounded) defaults of the form $P1 \wedge \ldots \wedge Pn : J1, J2, \ldots Jm / C$, such that $n+m=k$ and $Pi=SUBST(\theta, Xj)$ if $H \cup R \vdash SUBST(\theta, Xj)$, $Ji=SUBST(\theta, Xj)$ if $H \cup R \nvdash SUBST(\theta, Xj)$, and finally $C=SUBST(\theta, Y)$.

Our first proposal for the dynamic DfT construction presented in [14], is computationally unacceptable, since it requires that an attempts to prove literals from its knowledge base, in order to decide whether to use them in the prerequisite or the justification part of each default that it constructs; in other words, the agent needs to attempt to prove literals (and fail in doing so) in order to determine which of these are candidate assumptions.

In order to overcome this limitation we describe, in the next section, an alternative procedure by which an agent may determine assumptions dynamically and consequently construct the DfT. This technique does not require the agent to prove literals from its current knowledge base, and is, therefore, suitable for implementation.

3 Default Theory Construction and Inference

3.1 Rule Mapping

One may think of the 2^k possible defaults for one contract rule as organized in a triangle structure such as the one shown in Figure 1[2]. Each level of this triangle contains one or more of the 2^k defaults, depending on the number of assumptions that these defaults employ. That is, level 0 contains the single justification-free default, level 1 contains the k one-justification defaults, and so on, until the top level which contains the single, prerequisite-free default.

To illustrate this idea consider the following rule, which involves three conditions:

Y←X1∧X2∧X3

The corresponding 4-level triangle is:

Level 0: { X1∧X2∧X3 : true / Y }

Level 1: { X1∧X2 : X3 / Y, X1∧X3 : X2 / Y, X2∧X3 : X1 / Y }

Level 2: { X1 : X2, X3 / Y, X2 : X1, X3 / Y, X3 : X1, X2/ Y }

Level 3: { true : X1,X2,X3 / Y }

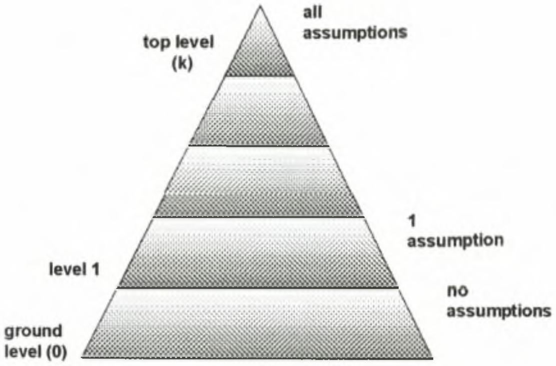

Figure 1 Triangle of assumptions

[2] For the moment we omit the normal default rule. We discuss normal defaults separately in section 3.3.

3.2 Inference Procedure

Of course, contracts (and normative systems in general) contain multiple rules, for each of which a triangle, such as the one described above may be constructed. All the resulting triangles are composed into a single polygon (Figure 2), which contains as many levels as the tallest of the constituent triangles (number of polygon levels=max(ki) where $1 \le i \le r$ and r is the number of contract rules). Thus, the DfT e-contract representation is also a pair of the form (W, D), where W is considered as already shown, and D contains the triangles of defaults that correspond to each initial contract rule. Note that, although the corresponding rule mapping is *one-to-many*, only one default for each initial contract rule may finally be employed for inference.

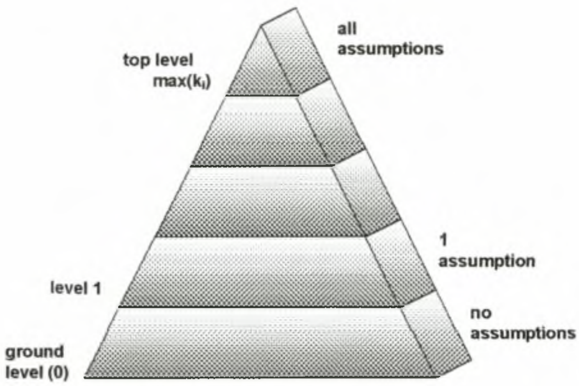

Figure 2 Polygon of assumptions

The polygon may be constructed either off-line, in advance, before an agent starts its inference, or incrementally during the inference process. In the first case appropriate computational and storage resources are needed in order to derive and store all the corresponding defaults for each initial contract rule. On the other hand, the gradual construction of each triangle, and consequently the gradual construction of the assumptions polygon, needs no particular resources, and thus a computational implementation is feasible.

The inference process starts from the ground level, by applying as many defaults as possible given the agent's current knowledge. Each time a default applies its conclusions are included in the current extension that is being computed. If the polygon has been constructed in advance, each time a default applies, the alternatives that lie in higher levels are ignored/removed from the polygon. If the polygon is constructed dynamically during the reasoning process, each time a default applies, the alternatives are not even computed and included in the higher levels. When there are no further defaults that can be applied in a level, this signals that assumptions are needed in order to proceed, and inference continues by examining defaults that lie in the next level upwards. Note that the case where reasoning is possible using only rules from the ground level is identical to inference in classical logic, but here we are also able to preserve consistency of entailment, if we want to employ appropriate variations of DfL such as Constrained Default Logic [35].

To illustrate the inference procedure, consider this next example: let us assume that a normative system comprises two rules of the form:

$R1=Y1 \leftarrow X1 \wedge X2$

$R2=Y2 \leftarrow X3 \wedge X4 \wedge X5$

Thus, the corresponding polygon levels contain the defaults:

Level 0: { $D1a0 = X1 \wedge X2 : true / Y1$, $D2a0 = X3 \wedge X4 \wedge X5 : true / Y2$ }

Level 1: { $D1a1 = X1 : X2 / Y1$, $D1b1 = X2 : X1 / Y1$, $D2a1 = X3 \wedge X4 : X5 / Y2$, $D2b1 = X3 \wedge X5 : X4 / Y2$, $D2c1 = X4 \wedge X5 : X3 / Y2$ }

Level 2: { $D1a2 = true : X1, X2 / Y1$, $D2a2 = X3 : X4, X5 / Y2$, $D2b2 = X4 : X3, X5 / Y2$, $D2c2 = X5 : X3, X4 / Y2$ }

Level 3: { $D2a3 = : X3, X4, X5 / Y2$ }

Here are some possible scenarios, with different initial knowledge available each time, in the beginning of the reasoning process:

- if $W=\{X1, X2\}$ then extension $In(2)=\{Y1, Y2\}$ is computed by making the assumption that X3, X4 and X5 hold ($Out(2)=\{\neg X3, \neg X4, \neg X5\}$) and by applying defaults D1a0 and D2a3 respectively.
- if $W=\{X1, X2, X3\}$ then extension $In(2)=\{Y1, Y2\}$ is computed by making the assumption that X4 and X5 hold ($Out(2)=\{\neg X4, \neg X5\}$) and by applying defaults D1a0 and D2a2 respectively.
- if $W=\{X1, X3, X4, X5\}$ then extension $In(2)=\{Y1, Y2\}$ is computed by making the assumption that only X2 holds ($Out(2)=\{\neg X2\}$) and by applying defaults D2a0 and D1a1 respectively.

Note that although a level may contain two or more defaults that correspond to the same initial contract rule (e.g. D2a1 or D2b1 or D2c1) there is no need for some kind of prioritization among those defaults. If two or more defaults of the same level, which are derived from the same initial rule, were to apply simultaneously, then the more general default contained in the immediately lower level should have applied.

We should note that it is important to consider the issue of consistency between assumptions employed during the reasoning process and new inferences derived as a result of the reasoning process. One of the reasons for which we revised our initial proposal for the construction of the DfT (which was described in section 2), is precisely because it would require a revision mechanism in order to reconstruct the default rules as new information becomes available, and the agent is able to prove literals from its updated knowledge, and hence treat them as pre-requisites rather than justifications. The alternative way that we propose here, for the construction of the DfT does not require any revision of the defaults. This is because inference involves one polygon level at a time in a step-wise manner, which ensures that the agent employs the fewest possible hypotheses.

Finally, note that the technique described here resembles, in a way, stratification of a DfT [5]. A DfT is stratified (SDfT) iff there exists a stratification function s that assigns a natural number to each default and, thus, separates the initial set of defaults D into strata. The stratification function is chosen so that, if the consequent of a default D1 is required as a pre-requisite or justification by another default D2, than D1 is to be applied before D2 i.e., $s(D1) \leq s(D2)$. Our separation of the possible set of defaults that correspond to each rule of the initial representation into levels, based on the number of assumptions employed, may be regarded as somewhat similar to a stratification criterion. We believe that it is worth examining the use of stratification, in its original sense, in combination with our proposed separation of the set of defaults based on the number of assumptions employed, to establish whether an agent's reasoning may be guided more thoroughly.

3.3 Normal Defaults

So far, we have omitted normal defaults from the discussion about the way in which an agent may construct its default theory. Normal defaults have the form $P:C/C$, i.e., their justification coincides with their consequent. Two questions seem to arise naturally:

(i) Should the agent include normal defaults in the set of potential mappings that it constructs from the initial e-contract representation? And, if so,

(ii) In which level of the triangle should normal defaults be placed?

It seems to us that normal defaults are required only in order to ensure that there is at least one extension of the currently available knowledge, which may be computed by adding to it new information, provided that consistency is preserved. That is, the normal default may be viewed as behaving similarly to the justification-free default, in that all its prerequisites should be satisfied by the current knowledge base; the only additional assumption made in the case of the normal default concerns the consistency of its conclusion with the current knowledge base. For this reason, although the normal default contains a single assumption, and should therefore belong to level 1 of the triangle, 'operationally' it belongs to level 0, since its assumption is not genuinely about something that holds in the world.

Hence, an agent may either omit normal defaults totally from the triangles that it constructs, or it may include them in level 0, if it is important to ensure that at least one extension exists while preserving consistency.

4 Prototype Implementation and Example

We found it useful to implement a prototype of the presented technique in Prolog for experimentation. So far, the prototype follows the specifications listed below:

- E-contracts are initially represented using propositional logic, and the tool constructs propositional DfTs.

- E-contract rules are represented as sentences of the form (1).
- Extensions are computed in the manner presented in [3], i.e., by maintaining syntactically consistent sets (In and Out) of formulae.
- Normal defaults are not considered.
- The assumptions polygon is constructed incrementally during the inference process.
- The applicability of defaults is checked in the same order that initial rules are given.
- In levels where that contain more than one corresponding defaults for the same initial rule, i.e. level 1 to level $\max(k_i)-1$ $(1 \leq i \leq r)$, the applicabilty of defaults is checked in the order that the defaults are placed in this level.

For the purposes of illustration consider a 3-party business transaction that takes place in an electronic marketplace populated by software agents. A buyer agent (BA) communicates with a seller agent (SA) and establishes an agreement with it for purchasing a certain product. Consequently, SA communicates with a carrier agent (CA) and establishes another agreement with it for the timely and safe delivery of goods to BA. An extract of the initial set of contract norms for the agreement between BA and SA is as follows[3]:

R={

R1= SAIsObligedToDeliverToBAWithinNext20days ←
BAOrdersFromSA
∧ E-shopFunctionsWell,

R2= BAIsObligedToPayCAOnBehalfOfBA ←
BAOrdersFromSA ∧ CADeliversToBA
∧ CAIsEmpoweredToAcceptPaymentFromBAOnBehalfOfSA

}

Note that these norms have the same form (number of conditions) as the norms considered in the example presented in section 3.2. Thus, the corresponding polygon coincides with the polygon computed in section 3.2. If W={ BAOrdersFromSA, E-shopFunctionsWell }, then the buyer may only infer, based on actual knowledge, that the seller is obliged to deliver products within the next 20 days. But there are cases where BA needs to perform:

- ***best-guess reasoning*** i.e., the agent is able to plan its ***future*** activities on the assumption that certain events/actions will occur, and that its partners' actions will be valid. For instance, consider that BA has just placed an order towards SA, i.e.

  W={ BAOrdersFromSA, E-shopFunctionsWell,
  CAIsEmpoweredToAcceptPaymentFromBAOnBehalfOfSA },

  and needs to plan its future activities (e.g. to infer the time of payment) assuming that all goes well and it receives the goods in due time. To derive such

[3] Note that this is simplified for the purposes of exposition here; the 'real' example is represented in Event Calculus, with actions/events, time points and fluents.

an answer BA needs to perform best-guess reasoning by employing the assumptions that delivery happens (CADeliversToBA) at some time point.

- *no-risk reasoning*, i.e., even though it may not know everything about the *past* and *present* the agent is able to derive a conclusion even though this is based on assumptions, because alternatively it might find itself in an undesirable situation. For instance, consider that delivery happens, i.e.

W={ BAOrdersFromSA, E-shopFunctionsWell, CADeliversToBA },

but no explicit knowledge is available about the CA's legal power to accept payment on behalf of SA. In this case, in order to avoid any extra charges, BA should employ CA's attribute (CAIsEmpoweredToAccept PaymentFromBAOnBehalfOfSA) and infer its obligation to pay.

5 Related Work

During the past twenty years various approaches and frameworks were proposed for assumption-based or hypothetical reasoning. These can be broadly grouped into:

(i) those that rely on a priori specification of assumptions that can be employed during the reasoning process, that is those that employ static assumptions; and

(ii) those that rely on ad hoc identification of potentially useful assumptions during the reasoning process, that is those that employ dynamic assumptions.

5.1 Static Assumptions

Doyle in 1979 [11] described the representation and structure of a Truth Maintenance System (TMS). As argued, this work solves part of the belief revision problem and provides a mechanism for making assumptions. This work is guided by the so called *problem of control* that is the problem of deciding on what will be the system's next inference. In other words, the agent needs an inference about which inference to make. New inferences are made by the Reasoner System (or overall Problem Solver) based on different assumptions that are statements believed without a particular reason. Consequently, different assumptions define different justified beliefs or reasoned arguments. A Truth Maintenance System, firstly, works as a cache by storing all inferences (justifications) ever made and, secondly, it makes any necessary revisions in the current belief set when the justifications-set, i.e. a set of justifications that represent different reasons for accepting a belief, is altered either by removing or adding a justification. In cases where a contradiction arises, a procedure, called *reasoned retraction of assumptions*, is introduced. The procedure searches on each belief justification-set for at least one assumption to be removed or added in order to eliminate the contradiction. In 1986, de Kleer in [8, 9] presented a new kind of TMS that avoids certain previous pitfalls. Contrary to [11] this new approach, the so called Assumption-based Truth Maintenance System (ATMS), is based on manipulating not only justifications but assumptions too. In this way, each belief is labeled with the set of assumptions under which it holds, besides the justifications that support it. Later, in [34] and [10] respectively, Reiter and de Kleer proposed some extensions and generalizations of the ATMS that are concerned mainly with the way the system is able to manipulate clauses more general than Horn clauses. Based on the above ideas of TMS and ATMS, Kohals *et al.* in [18, 2] proposed an extension of the propositional assumption-based model with probabilities, the so called Assumption-based Evidential Language (ABEL). Consequently, hypotheses were, also, enhanced with notions such as support, quasi-support, plausibility and doubt.

Poole in [28, 29] presents Theorist that is a framework for default reasoning implemented in Prolog. Poole argues that no special logic is required for default reasoning and proposes a modification to classical logic to achieve default reasoning. He considers the simplest case of hypothetical reasoning where the user provides the form of possible assumptions in order to achieve explanation. Specifically, Theorist accepts from users a set of closed formulae called facts (F), and a set Δ of potential assumptions called possible hypotheses. A closed formula is explainable from F and Δ if there is a set D of ground instances of Δ such that $F \cup D$ entail φ, and $F \cup D$ is consistent[4]. Finally, in [30] a very interesting discussion is presented. Queries such as "What are the possible hypotheses?" and "Who makes the assumptions?" are answered based on the type of problem the agent faces, i.e. planning, diagnosis or default reasoning.

Bondarenko *et al.* in [4] proposed an argumentation-based approach to hypothetical reasoning. This work is inspired by Dung's general argumentation framework and, specifically, it is based on the notions of ***attack*** and ***counterattack*** of the Argumentation Theory. An assumption is said to be acceptable if it is able to counterattack any other attacking set of assumptions. According to this view, definitions for admissible, complete, grounded, stable and preferred sets of assumptions were given. This fixed-assumptions framework is first introduced for logic programming, while an extension for its application to other formalisms of nonmonotonic reasoning is possible. Note that our previous comment about the EC representation viewed as a Logic Program with stable model semantics apply, here, also.

Kowalski and Sadri in [19, 20] compared the Situation Calculus (SC) [24, 33]] and the Event Calculus. Both calculi are formulated as Logic Programs. As noted, the EC was intended primarily for reasoning about actual events and the SC was primarily designed for reasoning about hypothetical actions. Thus the unification of the way both calculi handle hypothetical and actual events is proposed. Actual events are simply asserted in the knowledge base and their effects are considered valid. On the contrary, hypothetical events are also asserted in the knowledge base but nothing on their effects is stated. During the procedure of the assertion of events, integrity verification of the knowledge base is imperative. Integrity constraints are used to ensure that i) an event that happens is a possible event in the current situation and all its preconditions actually or hypothetically hold, and ii) no concurrent events are possible. Those constraints have a different role when dealing with actual or hypothetical events. In the first case constraints ensure that only possible events happen and in the second case constraints denote the context in which an assumption is possible.

[4] As Poole points out, his assumptions are identical to Reiter's supernormal default rules.

Provetti in [31] also deals with the problem of actual and hypothetical actions in terms of the Situation Calculus and the Event Calculus. Contrary to the Kowalski's and Sadri's approach, that unifies both calculi, Provetti introduces i) new predicates such as HypHolds(fluent,situation) to denote that a fluent is true in a situation, and ii) new ordered types of constants for denoting dates and functions denoting situations. A simple version of the EC formulated as an Extended Logic Program with answer sets semantics is presented and discussed as a tool for making assumptions on domains. Thus the new axiomatization of the EC is enhanced with new predicates and constants of the language.

Florea in [12] presents an assumption-based reasoning approach for multi-agent systems that is based on the TLI (Teoria Logica Implicita) logic. The proposed logic is a first order logic enhanced with special notations that describe Reiter's original default rules and help to derive extensions. In this work, the notion of the assumption coincides with Reiter's original notion of assumption.

Tahara in [37] addresses the issue of inconsistency that arises in the knowledge base due to inconsistent hypotheses. In this work, different contradictory scenarios, comprising of facts and hypotheses, are formed based on different hypothesis sets. Contradictions may be overcome using a preference relation between hypotheses. Thus, a scenario is represented as a triple $(F, H, <)$, where F denotes the set of facts, H denotes the set of hypothesis and $<$ denotes the partial ordered preference relation that holds among hypotheses.

All approaches discussed up to this point consider that hypotheses/assumptions and their preference relations, if any exist, are known a priori and provided by the software engineer or the user.

5.2 Dynamic Assumptions

Cox and Pietrzykowski in [7] explore the problem of the derivation of hypotheses to explain observed events, which is equivalent to finding what assumptions together with some axioms imply a given formula. They provide a method for computing causes of events that is based on linear resolution [23] and reverse Skolemization [6]. More importantly, this work studies and applies some restrictions that guarantee that the derived assumptions are in some sense interesting for our causing events. A cause of an event is: i) minimal, ii) consistent with the knowledge base, iii) nontrivial in the sense that $cause \supset event$ does not hold, and finally iv) basic iff every consistent cause of cause is trivial. Although, in [32] a top down search procedure, that is based on linear resolution, is described as a default proof procedure to explain a given wff, currently this view is out of our scope.

Abe in [1], also, deals with the problem of missing hypotheses when the system's aim is to explain an observation. He proposed a way to generate analogous hypotheses from the knowledge base when the latter lacks the necessary ones. This work adopts the previous work of Reiter and de Kleer [34] as a tool for abduction, called Clause Management System (CMS). A CMS, given an observation O that cannot be explained from the knowledge base KB ($KB \not\models O$), returns as set of minimal clauses O' such that $KB \models O \cup O'$ and $KB \not\models O'$. That is to say, O' is the minimal support for O with respect to KB, iff no proper subset of O' is support for O with respect to KB. The basic idea for hypothesis generation comprises two distinct steps: i) using first abduction and then deduction, candidate hypotheses are searched, and ii) in case where those candidate assumptions do not exist in the knowledge base, analogous hypotheses are being generated by referring to clauses in the knowledge base and to the result of the previous step. This work generates in an ad hoc manner the needed hypotheses during the inference, based on the ***relationship*** between clauses. Although this may seem a promising approach, we believe that the notion of analogy between clauses may lead us to unacceptable situations: consider that RA is obliged to perform payment by time point T. Thus RA performs payment via cash deposit in WA's bank account. But, what is the case if this way of payment is not an acceptable one? Although the action of paying via a deposit is analogous to the action of paying in cash their effects may differ.

Pellier and Fiorino in [26, 27] introduce a new approach to planning called Assumption-based Planning. Although, this approach is close to our approach regarding to the derivation of assumptions this work addresses the problem of hypothetical reasoning from a different perspective, i.e. goal achievement. Specifically, in [26] a mechanism that allows an agent to produce "reasonable" proposals according to its knowledge is presented. In this work, the meaning of the word "reasonable" is: i) goals cannot be considered achieved when they are based on conjectures and ii) the assumptions must be as few as possible. Actions' preconditions that cannot be proved are considered as conjectures, i.e. as additional goals to be satisfied in order to achieve the primary ones. They distinguish two kinds as assumptions: i) *hypotheses* that are literals that do not belong to the current knowledge base, and ii) *fact negations* that are the negation of literals replacing facts that an agent believes. The planning mechanism is based on the Hierarchical Transition Network (HTN) [25] where an agent decomposes *non-primitive* tasks into smaller subtasks until *primitive* tasks are reached, but unlike HTN a branch and bound algorithm is used in order to compute as few conjectures as possible.

Jago in [15] uses the notion of context in making assumptions. A context is the current set of the agent beliefs. This work argues that contexts are a suitable tool for modeling assumptions made within assumptions. Moreover contexts are also used as a tool in order to perform step-by-step temporal reasoning by considering that discrete contexts denote time points and, therefore, a sequence of contexts denotes a sequence of time points. Assumptions are not identified on an a priori basis and specifically, they are chosen either by guessing or are goal-driven.

Stamate in [36] presented a different approach to assumption-based reasoning. This work is close to our first approach regarding to the derivation of assumptions, but here a three-valued logic (i.e. *true*, *false* and *unknown* are the logical values for atoms) is adopted to express uncertainty in logic programs. Uncertainty is not only related to uncertain information but also to missing information, which is knowledge that is not derivable using the current knowledge and program rules. Consequently, a *pessimistic* assumption is made whenever underivable atoms are considered to be false. This case is identical to the CWA. A *skeptical* assumption is made whenever underivable atoms are considered to be unknown. And finally an *optimistic* assumption is made whenever underivable atoms are considered to be true.

6 Conclusions and Future Work

The work in this report is motivated from the need for assumption-based reasoning in an e-commerce setting, which is the application area of our project, and more generally in normative systems, where realistically agents will have incomplete knowledge about their environment, and about other agents. The question we seek to address is whether it is possible for agents to identify appropriate assumptions dynamically. In our previous work [13, 14], we argued that e-contracts could be represented as default theories and proposed a theoretical way in which such theories could be constructed automatically from initial Event Calculus representations. That proposal relied on determining what information could be proved from the agent's knowledge base, in order to decide whether it would serve as an assumption or not. In this report we propose an incremental technique that can be used for this construction that enables the dynamic and ad hoc identification of candidate assumptions without resorting to proof. We have developed a prototype implementation based on this idea, which translates initial propositional representations into propositional Default Theories.

Naturally we are interested in extending our implementation so that it may translate FOL representations into Default Theories. Note that in an initial (FOL) Event Calculus representation, all variables are implicitly assumed to be universally quantified. So the question that arises for the translation is what is the appropriate quantification for the variables that appear in the resulting default rules. There are four major approaches (cf. [32, 22, 29, 16, 17]) to the semantics of open default theories, and we have yet to investigate which one might be appropriate for computational purposes.

Acknowledgments

This work was supported by the European Social Fund (75%) and the Greek Secretariat for Research and Technology (25%) through the 3rd Community Support Programme - Measure 8.3 (PENED2003-03EΔ466).

References

[1] Akinori Abe. Two-sided hypotheses generation for abductive analogical reasoning. In *Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'99)*, pages 145–152, Washington, DC, USA, 1999. IEEE Computer Society.

[2] Bernhard Anrig, Rolf Haenni, Jörg Kohlas, and Norbert Lehmann. Assumption-based modeling using ABEL. In Dov M. Gabbay, Rudolf Kruse, Andreas Nonnengart, and Hans Jörgen Ohlbach, editors, *ECSQARU-FAPR*, volume 1244 of *Lecture Notes in Computer Science*, pages 171–182. Springer, 1997.

[3] Grigoris Antoniou. A tutorial on default logics. *ACM Computer Surveys*, 31(4):337–359, 1999.

[4] Andrei Bondarenko, Francesca Toni, and Robert A. Kowalski. An assumption-based framework for non-monotonic reasoning. In *LPNMR*, pages 171–189, 1993.

[5] Pawel Cholewinski. Stratified default theories. In Leszek Pacholski and Jerzy Tiuryn, editors, *CSL*, volume 933 of *Lecture Notes in Computer Science*, pages 456–470. Springer, 1994.

[6] Philip T. Cox and Tomasz Pietrzykowski. A complete, nonredundant algorithm for reversed skolemization. *Theor. Comput. Sci.*, 28:239–261, 1984.

[7] Philip T. Cox and Tomasz Pietrzykowski. Causes for events: Their computation and applications. In Jφrg H. Siekmann, editor, *CADE*, volume 230 of *Lecture Notes in Computer Science*, pages 608–621, New York, NY, USA, 1986. Springer-Verlag New York, Inc.

[8] Johan de Kleer. An assumption-based tms. *Artif. Intell.*, 28(2):127–162, 1986.

[9] Johan de Kleer. Extending the ATMS. *Artif. Intell.*, 28(2):163–196, 1986.

[10] Johan de Kleer. A general labeling algorithm for Assumption-based Truth Maintenance. In *Proceedings of the 7th National Conference on Artificial Intelligence. St. Paul, MN, August 21-26, AAAI Press / The MIT Press*, pages 188–192, 1988.

[11] Jon Doyle. A truth maintenance system. *Artif. Intell.*, 12(3):231–272, 1979.

[12] Adina Magda Florea. Assumption-based reasoning of intelligent agents. In *11th International Conference on Control Systems and Computer Science (CSCS'97)*, pages 126–133, 1997.

[13] Georgios K. Giannikis and Aspassia Daskalopulu. Defeasible reasoning with e-contracts. In *IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pages 690–694, Hong Kong, China, 2006. IEEE Computer Society.

[14] Georgios K. Giannikis and Aspassia Daskalopulu. The representation of e-Contracts as default theories. In *Proceedings of 19th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2007, accepted for publication*, 2007.

[15] Mark Jago. Modelling assumption-based reasoning using contexts. In *Workshop on Context Representation and Reasoning (CRR'05)*, 2005.

[16] Michael Kaminski. A comparative study of open default theories. *Artif. Intell.*, 77(2):285–319, 1995.

[17] Michael Kaminski, Johann A. Makowsky, and Michael L. Tiomkin. Extensions for open default theories via the domain closure assumption. *J. Log. Comput.*, 8(2):169–187, 1998.

[18] Jörg Kohlas and Paul-Andrn Monney. Probabilistic assumption-based reasoning. In David Heckerman and E. H. Mamdani, editors, *UAI*, pages 485–491. Morgan Kaufmann, 1993.

[19] Robert A. Kowalski and Fariba Sadri. The situation calculus and event calculus compared. In *Maurice Bruynooghe (Ed.): Logic Programming, Proceedings of the 1994 International Symposium, November 13-17, MIT Press*, pages 539–553, 1994.

[20] Robert A. Kowalski and Fariba Sadri. Reconciling the event calculus with the situation calculus. *J. Log. Program.*, 31(1-3):39–58, 1997.

[21] Robert A. Kowalski and Marek J. Sergot. A logic-based calculus of events. *New Generation Comput.*, 4(1):67–95, 1986.

[22] Vladimir Lifschitz. On open defaults. In J. Lloyd, editor, *Proceedings of the symposium on computational logic*, pages 80–95. Berlin: Springer-Verlag, 1990.

[23] Donald W. Loveland. A linear format for resolution. *Lecture Notes in Mathematics 125, Springer-Verlag, Berlin*, pages 147 – 162, 1970.

[24] John McCarthy. Situations, actions as causal laws. *Memo 2, Stanford University, Artificial Intelligence Project, Stanford, California*, 1963.

[25] Dana S. Nau, Tsz-Chiu Au, Okhtay Ilghami, Ugur Kuter, J. William Murdock, Dan Wu, and Fusun Yaman. Shop2: An htn planning system. *J. Artif. Intell. Res. (JAIR)*, 20:379–404, 2003.

[26] Damien Pellier and Humbert Fiorino. Assumption-based planning. In *AISTA'04: International Conference on Advances in Intelligence Systems Theory and Applications*, pages 367–376, Luxembourg-Kirchberg, Luxembourg, November 2004.

[27] Damien Pellier and Humbert Fiorino. Multi-agent assumption-based planning. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *IJCAI*, pages 1717–1718. Professional Book Center, 2005.

[28] D. Poole, R. Goebel, and R. Aleliunas. *Theorist: a logical reasoning system for defaults and diagnosis*, pages 331–352. Knowledge Frontier: Essays in the Representation of Knowledge, Springer Verlag, New York, 1987.

[29] David Poole. A logical framework for default reasoning. *Artif. Intell.*, 36(1):27–47, 1988.

[30] David Poole. *Who chooses the assumptions?* Abductive Reasoning, Cambridge: MIT Press., 1996.

[31] Alessandro Provetti. Hypothetical reasoning about actions: From situation calculus to event calculus. *Computational Intelligence*, 12:478–498, 1996.

[32] Raymond Reiter. A logic for default reasoning. *Artif. Intell.*, 13(1-2):81–132, 1980.

[33] Raymond Reiter. Proving properties of states in the situation calculus. volume 64, pages 337–351, 1993.

[34] Raymond Reiter and Johan de Kleer. Foundations of assumption-based truth maintenance systems: Preliminary report. In *Proceedings AAAI-87 Sixth National Conference on Artificial Intelligence, July 13-17, Seattle, WA. AAAI Press*, pages 183–189, 1987.

[35] Torsten Schaub. On constrained default theories. In *17th European Conference on Artificial Intelligence*, pages 304–308, 1992.

[36] Daniel Stamate. Assumption based multi-valued semantics for extended logic programs. In *ISMVL*, page 10. IEEE Computer Society, 2006.

[37] Ikuo Tahara. Computing scenario from knowledge with preferentially ordered hypotheses. *Systems and Computers in Japan*, 35(4):19–26, 2004.