5

Autonomous hypothetical reasoning: the case for open-minded agents

Aspassia Daskalopulu and Georgios K. Giannikis
University of Thessaly
Greece

1. Introduction

This chapter is concerned with *intelligent* software agents that populate *open* computational environments, in which they *interact* for various purposes, and in various manners, e.g. competitively in the case of electronic auctions or resource allocation problems, collaboratively in the case of distributed problem solving, parallel processing, joint planning, etc. By the term 'intelligent' we refer to agents that are *autonomous* (i.e. they decide for themselves what action to perform in order to meet their design goals), *rational* (i.e. they choose the best available course of action in order to meet their design goals; more precisely, they are *computationally rational*, in that they base their decision on the information available to them at the time of decision-making), and *social* (i.e. their interaction goes beyond mere data exchange and resembles social interaction among humans, that is, they may exchange promises, enter negotiations, raise requests, make demands, and so on). We use the term 'open' to characterize a computational environment in the sense defined by Hewitt (Hewitt, 1985), that is, in order to refer to an environment that is dynamic, continuous, unobservable (or, at best, partially observable) and non-deterministic. The interactions among agents in any multi-agent system are typically governed by *norms* that regulate the behaviour of the agents in the specific environment. Norms prescribe what behaviours are socially acceptable within a particular context, that is they specify what actions are obligatory, permitted or forbidden for each agent, in various circumstances, and usually within associated time bounds. For instance, norms may be used to regulate the agents' communication and coordination, and to specify liveness and safety properties of each agent, as well as the whole multi-agent system. In some cases, for example in electronic auction markets, or an information grid to which agents subscribe, these norms are designed *a priori* by the market owner, and when an agent joins the particular forum, this is taken to signal tacitly that the agent agrees to be subject to the market 'contract' that is specified by these norms. In other application areas, such as e-commerce exchanges or ad hoc distributed problem solving, and ad hoc task and resource allocation problems, agents may negotiate and agree between themselves the norms that will govern their interaction; by agreeing on a set of norms, agents signal tacitly that they agree to be subject to the 'contract' that is specified by these norms. However, since agents are autonomous and rational, and since the environment is open, their actual behaviour may deviate from the ideal behaviour prescribed by the

'contract', whether intentionally or unintentionally. For example, an agent that is obliged to perform a particular calculation at some specific time point, and return the result to some other agent, in a distributed problem solving scenario, may fail to comply with its obligation because at the designated time the agent lacked the computational resources to perform the calculation, or because by the time the agent completed the computation, it lost its communication means and could not deliver the result to the intended recipient; in an e-commerce scenario, a provider agent that is obliged to deliver a specific service or goods to a consumer agent may fail to do so, because it chose to deliver the service or goods to some other agent instead, after it established that the income that it would receive in this way outweighed the reparation costs that it would owe to the consumer agent. Autonomous agents decide for themselves what actions to perform, as noted earlier, and this decision includes the decision on whether to comply with the 'contract' that governs their behaviour. An agent's rationality is measured against the quality of its decisions, and this relies heavily on the quantity and quality of information that the agent possesses at the time point of decision-making. Unavoidably, agents is open environments possess information that is incomplete, imprecise, maybe even incorrect, due to the very fact that the environment is open and, at the very least, agents join and leave it as they choose. Information exchange between agents may be delayed, and message content may be distorted by noise during communication; it may even be intentionally false in the case of insincere agents. It is natural, therefore, to expect that agents will have to perform their decision-making by employing assumptions, in order to fill in what they perceive as information gaps. Assumption identification and deployment must be dynamic, since the agents operate in a dynamic environment, and the agents' reasoning must be nonmonotonic, since any conclusions drawn on the basis of assumptions may need to be revised, should information that was previously missing become available later.

We begin by motivating the need for dynamic and autonomous hypothetical reasoning, and we identify and state explicitly three aspects of this problem in the context of open norm-governed environments. Then we present our approach to dynamic assumption identification and usage, and we demonstrate the way in which it permits agents to establish their current knowledge state, as well as their current assumption requirements, autonomously. Our approach exploits the syntax of Default Logic (DfL) rules (Reiter, 1980), in order to separate the definite knowledge from the assumptions that are used in drawing a conclusion; however, as we explain later, assumption identification and deployment is conducted without resorting to proof, which is notably computationally hard. Finally, we review and discuss the main other approaches to assumption-based reasoning found in the Artificial Intelligence literature. We should note from the outset that the distinctive feature of our approach, which sets it apart from these other approaches, is that agents do not rely on a pre-specified pool of assumptions, in order to identify their assumption requirements. Nor do they rely on goal-orientation, as a means to identify candidate assumptions. In this way, we argue, an agent is *open-minded*, in that it decides for itself which assumptions are plausible and appropriate at any given time, and crucially, this involves making assumptions about its past, as well as its future; as it turns out it manages assumption deployment in a rational manner.

2. Who needs assumptions and when?

Rule-based knowledge representation, typically in first-order logic or some subset of it, has become popular and is used in many symbolic Artificial Intelligence applications, in order to encode domain information, as well as the decision-making behaviour of the system. Typically, rules have the form of sequent calculus sentences (Jean-Yves Girard and Lafont, 1989), i.e.

$$Y \leftarrow X_1, X_2, \ldots, X_k \quad (1)$$

where Y and Xi ($1 \leq i \leq k$) are positive or negative literals (any variables are assumed to be universally quantified) representing the rule conclusion and conditions, respectively. The semantics of such a rule is "*if all of the conditions Xi hold, then conclusion Y holds*". The system checks whether conditions hold against its knowledge base, and the conclusion Y it draws, in case the conditions are true given its knowledge, may result in belief revision – the knowledge base is updated to contain new information – or in the actualization of some behaviour – the system does something, and this action may be private (some internal computation), or public (e.g. the dispatch of a message, or, in the case of robotic agents, the performance of some action that transforms the environment in some way).

A question that arises naturally is: what happens when the system does not know all of the conditions Xi that are involved in some rule contained in its knowledge base, i.e. it does not have explicit information, in its knowledge base, about the truth or falsity of some, or all of the conditions of a rule? There are two options: (i) if the system employs the Closed World Assumption (CWA) (Reiter, 1977) - anything not explicitly known is considered false - then the unknown conditions are treated as false, and hence the rule does not apply; (ii) if the system employs the Open World Assumption (OWA) - anything not explicitly known is considered neither true, nor false, merely unknown - then the evaluation of unknown conditions is unsuccessful, and, again, the rule does not apply. In the first case the system deals with information gaps by filling them in, in a narrow-minded manner, based only on its set of beliefs; in the second case, the system is open-minded, in that it remains agnostic about missing information, allowing for the possibility that something it does not know may be true, but this stance is counterproductive, since the system cannot in effect apply its rule. In the worst case, if all of the rules of its knowledge base do not apply, because some of their conditions are unknown given the current knowledge base, the system will do nothing, whether it employs the CWA or the OWA. At best, in order to avoid idleness, the system may ask its user to determine the truth value of the information it lacks. This is, in fact, the classic approach to inference employed in expert systems, where the system user becomes responsible for the quality of information that is available to the system, and ultimately for the quality of the conclusions that the system draws. The user herself may not have definite knowledge about the missing information, yet she may want the system to proceed and produce even a tentative result; in this case the user provides an answer to the system, noting to herself that this answer is, in fact, an assumption that she makes for the time being, yet the system is not aware that it is employing an assumption, nor does it have any control over assumption deployment, since it is not up to it to decide whether to use the answer provided by the user or not.

This approach is clearly inappropriate for intelligent agents, which are, after all, designed and developed in order to perform tasks delegated to them by humans, and which by

definition, must behave autonomously and rationally. Typically, intelligent agents interact with other agents in a designated computational or physical environment, and this interaction is regulated by norms that stipulate what each agent is obliged, permitted, prohibited, institutionally empowered and so on to do, much in the same spirit that human agent interaction is regulated by the Law, organizational, and other formal, or informal, social rules. Human agents do not always comply with legal, organizational and social norms. The mere existence of these norms, whose purpose is to describe which behaviours are acceptable, encouraged, or even imperative, does not guarantee that their subjects will, in fact, behave accordingly. The actual behaviour that a human agent demonstrates deviates, quite often, from the ideal, and this may happen intentionally or unintentionally. It may be the case that a human agent knows what the norm prescribes, knows the consequences of a potential violation, but nonetheless chooses to violate it, for various reasons – she cannot do otherwise, or she judges that the penalty associated with the violation is worth paying, for the violation itself results in some state that she considers in some way positive, given her subjective value system. It may be the case that human agents violate norms unintentionally, often for the simple reason that they do not know that specific norms apply. In similar spirit, autonomous, rational, artificial agents cannot be expected to behave in accordance to the norms that govern their virtual societies, for the simple reason that they are designed and developed to act in the interest of their human/organizational owners, in accordance with the latter's value systems and goals.

Social norms may be thought of as rules of the form (1), shown above, where the conditions Xi and the conclusion Y refer not only to application domain entities, but to normative notions as well, which characterize agents' actions, or the states of affairs that can be effected by agents' actions; the main normative notions are obligation, permission, prohibition, and institutional power (the ability to create normative relations). For instance, in an electronic commerce scenario, a norm may specify that "*if the seller agent delivers the specified quantity of goods to the buyer agent by the due date, then the buyer agent is obliged to pay the specified amount to the seller agent by a specified date*"; in a distributed problem solving scenario, a norm may specify that "*if the planning agent computes a partial plan for a specified goal, then it is permitted for the planning agent to send the partial plan it computed to the coordinator agent immediately*", or that "*a planning agent is prohibited from sending partial solutions without being asked to do so by the coordinator agent*", and another may specify that "*the coordinator agent is prohibited from computing the final plan, unless it has received partial plans for all of the subgoals that it allocated to planning agents*"; in a task or resource allocation scenario a norm may specify that "*the scheduler agent is permitted to change the order of print jobs in the printer queue, when a print job arrives from a designated port, which is to be treated as an emergency port*", or another norm may specify that "*a scheduler agent is empowered to impose an obligation on a printer agent that processes jobs in first-in-first-out manner to start processing jobs in shortest-job-first manner, whenever the queue reaches a certain capacity*"

In order for an agent to decide whether to comply or not with a norm, first it must establish that the norm applies, and to do this it must be able to establish both factual information and prescriptive information, given a history of events that have occurred up to the point of its query. We saw earlier that a system that lacks information about rule conditions cannot apply its rules and, inevitably, if it is not to remain idle, it has to resort to its owner and obtain answers from her. This is not a realistic solution in the case of autonomous, rational agents, which are designed to act on behalf of their owners – idleness is not an attractive

option, and an autonomous agent is expected to turn to its owner for help only in extreme circumstances!

Obviously agents in any system (not necessarily open) do not possess information about the future. In order for an agent to meet its design goals, though, and plan its course of action at any given time, it needs to fill in information gaps by employing assumptions about the future. In open computational environments, even the historical information available to an agent when it poses its query may be incomplete, for various reasons: Information may be lost, or distorted by noise, and in a truly open system, where agents join or leave the system at different times, information delivery from agent to agent may simply be delayed. In order to reason in the absence of complete historical knowledge, agents must be able to fill in information gaps, by employing assumptions about the past and the present. One might argue that it would be reasonable for an agent to adopt the CWA in order to deal with information gaps that concern the past, and the OWA in order to deal with information gaps that concern the future. However, such an agent would not be truly open-minded, in that it would treat the historical information that it would possess as definitive. And in any case, as we argued earlier, both the adoption of the CWA and the adoption of the OWA have the same practical effect: the agent would remain idle, or it would need to resort to its owner, in order to fill in information gaps and proceed with its inferences.

Therefore assumption-based reasoning is useful in two modes, progressively, because the agent cannot know the future, and retrospectively, because the agent may not know the past. Progressive hypothetical reasoning is sometimes referred to as *best-guess reasoning*, while retrospective hypothetical reasoning is sometimes referred to as *no-risk reasoning*.

- *Best-guess reasoning*: An agent 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 or other agents' actions will occur, or that certain causal relations will be effected in the environment, or that it will bear a certain normative status (obligations, permissions, prohibitions, powers) towards other agents.
- *No-risk reasoning*: An agent may not know everything about the past and present, i.e., the history of its environment, other agents and itself so far, yet it may need to plan its activities on the basis of hypotheses that concern the past and present, i.e., on the assumption that certain events or other agents' actions have occurred, or that certain normative relations have obtained between itself and other agents, in order to protect itself from an undesirable situation in the future.

To illustrate these cases, consider a business transaction that takes place in an electronic marketplace populated by software agents. A buyer agent communicates, at time point T, with a seller agent and establishes an agreement with it for purchasing a certain product. Consequently, the seller agent communicates with a carrier agent and establishes another agreement with it for the timely and safe delivery of goods to the buyer agent. A reasonable query that the buyer agent might have might be general, such as "*What do I have to do from now on, with respect to this contract?*". The buyer agent would expect a list of all the obligations, permissions, prohibitions, etc. that arise for it as a result of entering this agreement, along with their associated times, which may be relatively or absolutely defined. However, such an answer can only be derived on a hypothetical basis, that is on the assumption that the buyer agent's order will be received correctly and on time by the seller agent, that the seller agent will acknowledge its obligation to respond to such an order with delivery, and that it will, in fact, be able and willing to provide the required quantity and

quality of goods in time, that the buyer agent will acknowledge its obligation to pay for the goods that it will receive, and so on. The buyer agent's query about the future might be more specific, such as "*When will I, potentially, have to pay for this order, assuming all goes well and I receive the goods in due time, so that I plan to have adequate available funds?*". To derive an answer the buyer, again, needs to reason on the basis of future hypotheses, i.e. to perform best-guess reasoning. Now, consider the case where, after placing an order at time T, the buyer agent at some subsequent time point T'' (T<T''), wonders "*I placed an order at time point T, and so far I have not received any information about how this order is proceeding. What if the seller agent has already dispatched the goods to me, and is expecting payment from me, while I am blissfully unaware that I should do something about this?*". In this case, the buyer agent does not know whether the carrier agent has performed delivery at some point T', such that T<T'<T''. Unless the buyer agent resorts to an assumption about the past (that delivery happened at some time point T'), that is unless it performs no-risk reasoning, it cannot infer that an obligation for it to pay the seller has become active, and it risks finding itself in the undesirable situation, where its time allowance for paying has expired and it now bears a sanction, say to pay some extra amount to the seller agent, to compensate for missing its deadline.

We see that the reasoning problem faced by an agent in this context involves the following aspects:

1H. *Assumption identification and usage*: What assumptions are applicable to fill in information gaps and how should these be employed in the inference process?

2H. *Assumption influence*: What is the relation between the assumptions and the current or future world, i.e. how do assumptions employed at a given time point enable or restrict present and future conclusions?

3H. *Assumption corroboration*: What happens when new information becomes available at some time point, confirming or disproving assumptions employed at earlier time points, or conclusions drawn at earlier time points?

In order to address question 1H an agent seeks to identify appropriate assumptions, and because it operates in an open, essentially dynamic, environment, assumption identification must be carried out dynamically. In order to answer question 2H the agent needs to employ some way that commits its reasoning to the specific assumptions that it employs, from that moment onwards. Finally, in order to answer question 3H the agent needs to reason nonmonotonically.

There are many interesting approaches to dynamic assumption-based reasoning, which we review in section 4, after we present our approach in section 3. We stress from the outset that these approaches rely either on the existence of a pre-specified assumption space or on pre-specified criteria for the identification of assumptions. In the first case, assumption identification is not really dynamic, rather assumption usage, i.e. the management of the pre-specified assumption space, may be done dynamically. In the second case assumption identification is dynamic, only in the sense that the appropriate assumption is chosen at run-time, but since this choice is made on pre-specified criteria, it is in a sense static. What distinguishes, therefore, our work from these approaches, is that we propose a way in which both identification and usage of appropriate candidate assumptions are done dynamically. In this way, we argue, the agent is truly autonomous in deciding what assumptions to employ and when to employ them. We start by presenting our approach, before we review

these other approaches, in order to facilitate the reader in appreciating these differences, without getting lost in technical detail.

3. Autonomous hypothetical reasoning

In (Giannikis and Daskalopulu, 2006, Giannikis and Daskalopulu, 2007) we proposed an approach for representing norms (focussing specifically on e-contracts), as Default Theories (DfT), constructed dynamically from an initial Event Calculus (EC) representation (Kowalski and Sergot, 1986).

Many researchers have explored the use of temporal logics for the representation of e-contracts (e.g. (Marvn and Sartor, 1999, Artikis *et al.,* 2002, Farrell *et al.,* 2005, Rouached *et al.,* 2005)), and have demonstrated how such representations allow us to establish the state of a business exchange, given the actions that parties perform or omit to perform. On the basis of such a representation and in order to support nonmonotonic reasoning, one might use some of the various approaches such as Circumscription (McCarthy, 1980), Logic Programs (Gelfond and Lifschitz, 1988, Gelfond and Lifschitz, 1991), or Defeasible Logic (Nute, 1994), as many other researchers have done. In (Giannikis and Daskalopulu, 2007) we discussed in detail our view on the extent to which these approaches can cope with hypothetical reasoning, and argued for the adoption of Default Logic for the following reasons:

(i) The syntax of DfL offers a natural way to represent separately what is known, what is assumed and what is concluded on the basis of this knowledge and assumptions. We saw an opportunity to exploit this syntax, in order to address the first aspect (1H) of our problem, namely assumption identification and usage.

(ii) The semantics of DfL offers a way to reason nonmonotonically and to preserve the relation of an assumption and any inferences drawn on its basis. This enables us to address the second and the third aspect (2H, 3H) of our problem, namely assumption influence and assumption corroboration.

However, we should note that although we exploit the syntax of DfL, we do not resort to proof in DfL, which is notably computationally hard. Instead we adapt an incremental technique for the computation of possible world models, initially proposed in (Antoniou, 1999) which requires set manipulation.

3.1 Preliminaries

A default rule (henceforth default) has the form:

$$P : J_1, J_2, \ldots J_n \,/\, C,$$

where P is the prerequisite, $J=\{J_1, J_2, \ldots J_n\}$ is a set of justifications, and C is the derived consequent of the rule. The semantics of this rule is: If P holds and the justifications contained in J are consistent with the current knowledge, then C may be inferred. 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 $E \supseteq W$, if and only if $P \in E$ and $\neg J_1 \notin E, \ldots, \neg J_n \notin E$. The set E is called the *extension* of the DfT. The notion of extension is the most complicated concept of Reiter's logic, because it is hard to determine an accurate belief set for which justifications

should be consistent. In his initial paper on DfL (Reiter, 1980) Reiter noted three important properties of extensions: (i) an extension E of a DfT (W, D) should contain W, (ii) the extension E of a DfT should be deductively closed, and (iii) for a default rule of the form $P:J_1,J_2,\ldots J_n / C$, if $P \in E$ and $\neg J_1,\ldots, \neg J_n \notin E$ then $C \in E$.

The requirement that the extension of a DfT be deductively closed is computationally problematic. However, Antoniou in (Antoniou, 1999) proposed a useful operational definition of extensions and a technique for their computation, which is done incrementally, by maintaining syntactically consistent sets of formulae, whose conditions part (prerequisites and justifications) is interpreted conjunctively and the conclusions part (consequent) is interpreted disjunctively, as in sequent calculus.

Let Π represent a default reasoning process by recording the order in which defaults from D apply. At each step i of the reasoning process, i.e. after the application of each default $P:J_1,J_2,\ldots,J_n/C$, the extension computed is a set of ground sentences $In(i)=In(i-1) \cup \{C\}$, and the set of justifications employed, which should not turn out to be true, is $Out(i) = Out(i-1) \cup \{\neg J_1\ldots, \neg J_n\}$. As a result, $\Pi(i)= \Pi(i-1) \cup \{Di \mid Di$ is the default rule which applied at step i$\}$. Initially $In(0)=W$, $Out(0)=\varnothing$ and $\Pi(0)=\varnothing$ for i=0. The default reasoning process $\Pi(i)$ is successful if and only if $In(i) \cap Out(i)= \varnothing$, otherwise it is failed. Moreover, the process $\Pi(i)$ is closed if and only if every default rule that belongs in the set D and is applicable to In(i) already occurs in $\Pi(i)$. According to (Antoniou, 1999) a set of formulae E is a DfT extension, if there is a closed and successful process $\Pi(i)$ of the DfT such that E=In(i).

For a quick illustration of these concepts, consider the DfT (W, D), where W={A} and D contains the following defaults:

$$D1 \equiv A : B / C$$

$$D2 \equiv true : \neg D / E$$

The process $\Pi(2)=\{D1,D2\}$, i.e. $In(2)=\{A,C,E\}$ and $Out(2)=\{\neg B,D\}$, is successful and closed, thus it is considered as an extension of the theory.

Now, consider the DfT (W, D), where W={A} and D contains the following defaults:

$$D1 \equiv A : B / C$$

$$D2 \equiv true : D / \neg B$$

The process $\Pi(2)=\{D1,D2\}$, i.e. $In(2)=\{A,C,\neg B\}$ and $Out(2)=\{\neg B,\neg D\}$, is closed but not successful, thus it is not considered as an extension of the theory. The process $\Pi(1)=\{D2\}$, i.e. $In(1)=\{A,\neg B\}$ and $Out(1)=\{\neg D\}$, is successful and closed, since D1 does not apply, thus it is considered as an extension of the theory.

3.2 The idea in a nutshell

We saw an opportunity to exploit the syntax of default rules in order to represent the inference relation between what the agent knows definitely (the premises in Reiter's terms), what the agent can assume, consistently with its current knowledge (the justifications in Reiter's terms), and what the agent can infer on the basis of its current knowledge and the assumptions that it adopts (the consequent in Reiter's terms). Therefore, the basic idea of our

approach, which was initially proposed in our (Giannikis and Daskalopulu, 2006), is to have agents reason, via constructing extensions of default theories, using Antoniou's incremental technique. To achieve this, the agents must reason with default theories, that is, their knowledge base, whose contents are initially in the form of sentences of sequent calculus, must be translated into default rules.

Recall, that initially an agent's knowledge base contains sentences of the form (1):

$$Y \leftarrow X_1, X_2, \ldots, X_k$$

where Y and Xi ($1 \leq i \leq k$) are positive or negative literals (any variables are assumed universally quantified) representing the rule conclusion and conditions, respectively. A rule of such a form must be translated into a default rule, where what the agent knows definitely will be captured by the prerequisite component, what the agent does not know will be captured by the justification component, and the conclusion Y will be captured by the consequent component. In principle the agent's definite knowledge changes over time, while it interacts with other agents and it perceives its environment, so in principle any one or all of the conditions Xi of such a rule may be known or unknown. Hence, a single rule of the form (1) shown above may be mapped to *any* one of the following default rules:

- To the single assumption-free default of the form $X_1, X_2, \ldots, X_k : \text{true} / Y$, just in case the agent possesses definite knowledge about all of the conditions Xi, and therefore needs to make no assumptions, in order to be able to apply the rule and derive a conclusion.
- To k one-assumption defaults, that correspond to the k possible situations, where the agent possesses definite knowledge about k-1 of the conditions Xi, and needs to make a single assumption for the unknown condition, that is to one of the defaults of the set:

$$\{ X_1, X_2, \ldots, X_{k-1} : X_k / Y,$$
$$X_1, X_2, \ldots, X_{k-2}, X_k : X_{k-1} / Y,$$
$$X_1, X_2, \ldots, X_{k-1}, X_k : X_{k-2} / Y,$$
$$\ldots$$
$$X_2, \ldots, X_k : X_1 / Y \ \}$$

- To k(k-1)/2 two-assumption defaults, that correspond to the k(k-1)/2 possible situations, where the agent possesses definite knowledge about (k-2) of the conditions Xi, and needs to make assumptions for the two unknown conditions, that is to one of the defaults of the set:

$$\{ X_1, X_2, \ldots, X_{k-2} : X_{k-1}, X_k / Y,$$
$$X_1, X_2, \ldots, X_{k-3}, X_{k-1} : X_{k-2}, X_k / Y,$$
$$X_1, X_2, \ldots, X_{k-4}, X_{k-2}, X_{k-1} : X_{k-3}, X_k / Y,$$
$$\ldots$$
$$X_3, \ldots, X_k : X_1, X_2 / Y \ \}$$

- In similar spirit, to any one of the set containing three-assumption defaults, four-assumption defaults and so on, right down to the single k-assumption default, which corresponds to the case where the agent knows nothing and has to make assumptions about everything, i.e. $\text{true} : X_1, \ldots, X_k / Y$.

So, each initial rule, which involves k conditions, may be translated into any one of 2^k defaults[1]. The question is, which one is the appropriate one? And the answer is, let the agent decide, depending on its current knowledge/absence of knowledge state, that is, let the agent determine what it knows and what it needs to make assumptions about, dynamically, as its knowledge base evolves over time.

Our first proposal for the translation of an agent's initial knowledge base into a DfT appeared in (Giannikis and Daskalopulu, 2007): Given a sequent calculus rule in its knowledge base, the agent would construct a default rule by mapping the conclusion of the rule to the consequent part of the default, all of the conditions Xi that could be proved from its knowledge base to the prerequisite part of the default, and the remaining conditions that could not be proved from its knowledge base to the justification part of the default. Although this is a correct formal characterization of the intended translation, it is computationally unacceptable, since it requires an agent to attempt 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 identify candidate assumptions. In order to overcome this limitation we proposed an alternative view in (Giannikis and Daskalopulu, 2008), which is suitable for implementation, and relies on structuring hierarchically the 2^k possible translations into a multi-level structure and have the agent traverse it. We present this in detail in the next section.

3.3 Default theory construction and inference

We may think of the 2^k possible defaults for a single rule of the form (1) as representations of the possible mental states in which the agent may find itself. Each such state is characterized by what is known and what is not known to the agent, i.e. it represents what we may call the single-rule knowledge/hypothesis (KH) status of the agent. These possible states are organized in a multi-level hierarchy, which we depict as a triangle, such as the one shown in Figure 1. The top of the triangle shows the direction in which the agent's mental state evolves over time. Each level of the KH structure contains those of the 2^k possible default translations of the rule that contain as many assumptions as indicated by the number of the leverl, that is level 0 contains the single assumption-free default, level 1 contains the k one-assumption defaults, and so on, until the top level which contains the single, knowledge-free default. That is, for a an agent which possesses an initial rule of the form (1), moving upwards in a stepwise manner until it reaches the top level of the single-norm KH structure, is tantamount to identifying candidate assumptions among the conditions that are included in the initial rule. Defaults contained in the same level have the same number of assumptions; the defaults of any given level contain one more assumption than the defaults of the immediately lower level, and one fewer assumption than the defaults of the

[1] To be precise, we should note that there is one more default that could be used as the translation of the initial sequent calculus rule, namely the normal default of the form $X_1, ..., X_k : Y / Y$, which corresponds to the case where the agent knows all of the conditions, and proceeds to infer Y, if assuming it is consistent with its current knowledge. There is a short philosophical discussion that can be made about this point, but we leave it aside for the time being, and we shall return to it in the discussion at the end of this section, after we have presented the details of our approach.

immediately higher level. Let | L | denote the total number of defaults contained at level L, where $0 \leq L \leq k$, and k is the total number of conditions in an initial rule of the form (1). Then, it is easy to verify that the following properties hold:

- $|L| = 1$ if $L = 0$
- $|L| = (k - L + 1) * |L-1| / L$ if $L \neq 0$

To illustrate this idea consider the following rule, given in the initial sequent calculus form, which involves four conditions (k=4):

$Y \leftarrow X_1, X_2, X_3, X_4$

The corresponding 5-level triangle is:

Level 0: { X_1,X_2,X_3,X_4 : true / Y }

Level 1: { $X_1,X_2,X_3 : X_4$ / Y,
$X_1,X_2,X_4 : X_3$ / Y,
$X_1,X_3,X_4 : X_2$ / Y,
$X_2,X_3,X_4 : X_1$ / Y }

Level 2: { $X_1,X_2 : X_4, X_3$ / Y,
$X_1,X_3 : X_4, X_2$ / Y,
$X_2,X_3 : X_4, X_1$ / Y,
$X_1,X_4 : X_3, X_2$ / Y,
$X_2,X_4 : X_3, X_1$ / Y,
$X_3,X_4 : X_2, X_1$ / Y }

Level 3: { $X_1 : X_4, X_3, X_2$ / Y,
$X_2 : X_4, X_3, X_1$ / Y,
$X_3 : X_4, X_2, X_1$ / Y,
$X_4 : X_3, X_2, X_1$ / Y }

Level 4: { true : X_4,X_3,X_2,X_1 / Y }

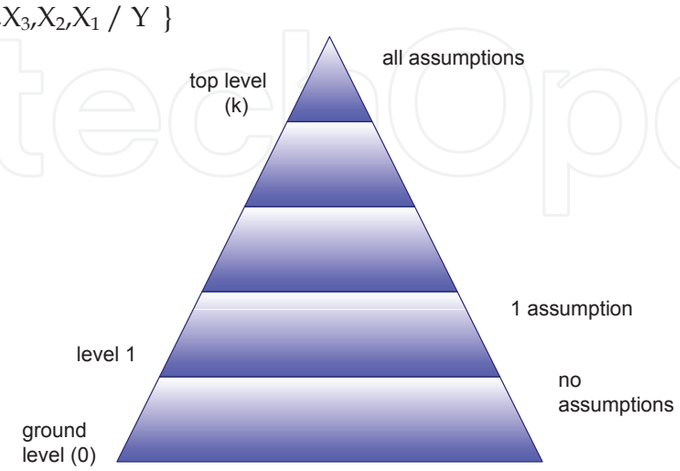

Fig. 1. Single-rule KH structure of an agent's mental states

An agent's initial knowledge base will typically contain many rules, for each of which the agent constructs a KH structure. All the resulting single-rule KH structures are composed into a single polygon-like structure (Figure 2), which contains as many levels as the tallest of the constituent single-rule KH structures. Given an initial set of rules, the number of levels of the multi-rule KH structure is equal to the maximum k_i, where $1 \leq i \leq r$ and r is the number of the initial norms of the form (1). To be precise, we should note that the multi-rule KH structure does not have a single top, since each constituent single-rule KH structure may have its own top level. We are interested in the highest amongst these top levels, since this denotes the point of termination of an agent's inference process, when an agent moves upwards in the multi-rule KH structure and its mental state evolves over time.
Therefore, the full DfT that is constructed by an agent is a pair of the form (W, D), where W contains all of the available (if any) historical information and D is the multi-rule KH structure. Level 0 contains the r assumption-free defaults, level 1 contains the $\sum^{r}_{i=1} k_i$ one-assumption defaults, and so on, until the top $\max(k_i)$ level, which contains some of the knowledge-free defaults.
Note that, although the corresponding rule mapping is one-to-many, only one default for each initial rule may finally be employed for inference. 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 consequent is included in the extension that is being computed currently. When there are no further defaults that can be applied in a level, this signals to the agent that assumptions are needed in order to proceed, and inference continues by examining defaults that lie in the next level upwards.

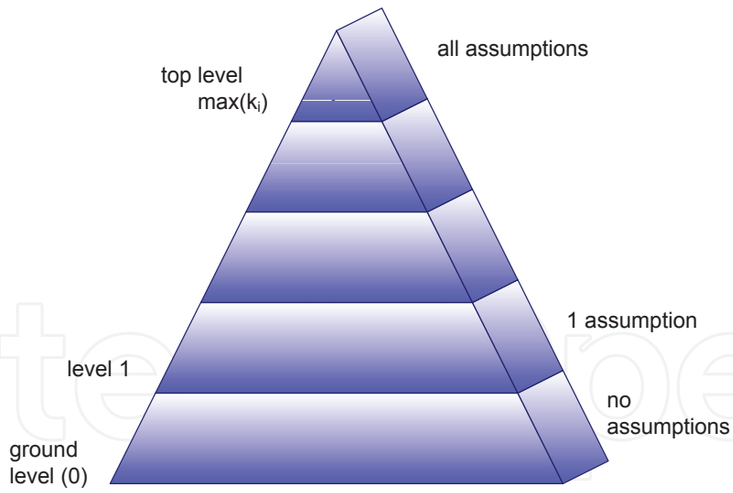

Fig. 2. Multi-rule KH structure of an agent's mental states

Note that the case where reasoning is possible using only rules from the ground level is identical to inference in classical logic. During its reasoning process an agent infers all possible conclusions on the basis of its current knowledge. When no further inference is possible, the agent is able to reassess its mental state and establish its knowledge/hypothesis status, in order to continue. That is, the agent first attempts to draw conclusions using only assumption-free defaults, then by employing one assumption per default, then by employing two assumptions per default, and so on, until no further defaults

apply. In other words, a general priority criterion among defaults is being established: This is the number of assumptions employed via the use of a default rule. Thus, such inference in a step-wise manner ensures that the agent employs the fewest possible hypotheses, always, that is, that the agent is rational in its deployment of assumptions.
To illustrate the inference procedure, consider this next example: let us assume that a normative system comprises two rules of the form:

$$R1 \equiv Y_1 \leftarrow X_1, X_2$$

$$R2 \equiv Y_2 \leftarrow X_3, X_4, X_5$$

Thus, the corresponding single-rule and multi-rule KH structures are as follows ($D_{level,number}$ denotes the level of the default and its identification number within its level, and it is used to facilitate reference):

Single-rule KH structure for R1:
Level 0: { $D1_{0,1} \equiv X_1,X_2 : true / Y_1$ }

Level 1: { $D1_{1,1} \equiv X_1 : X_2 / Y_1$,
$D1_{1,2} \equiv X_2 : X_1 / Y_1$ }

Level 2: { $D1_{2,1} \equiv true : X_2,X_1 / Y_1$ }

Single-rule KH structure for R2:
Level 0: { $D2_{0,1} \equiv X_3,X_4,X_5 : true / Y_2$ }

Level 1: { $D2_{1,1} \equiv X_3,X_4 : X_5 / Y_2$,
$D2_{1,2} \equiv X_3,X_5 : X_4 / Y_2$,
$D2_{1,3} \equiv X_4,X_5 : X_3 / Y_2$ }

Level 2: { $D2_{2,1} \equiv X_3 : X_5, X_4 / Y_2$,
$D2_{2,2} \equiv X_4 : X_5, X_3 / Y_2$,
$D2_{2,3} \equiv X_5 : X_4, X_3 / Y_2$ }

Level 3: { $D2_{3,1} \equiv true : X_5, X_4, X_3 / Y_2$ }

Multi-rule KH structure for R1 and R2:
Level 0: { $D1_{0,1} \equiv X_1,X_2 : true / Y_1$, $D2_{0,1} \equiv X_3,X_4,X_5 : true / Y_2$ }

Level 1: { $D1_{1,1} \equiv X_1 : X_2 / Y_1$, $D2_{1,1} \equiv X_3,X_4 : X_5 / Y_2$,
$D1_{1,2} \equiv X_2 : X_1 / Y_1$, $D2_{1,2} \equiv X_3,X_5 : X_4 / Y_2$,
$D2_{1,3} \equiv X_4,X_5 : X_3 / Y_2$ }

Level 2: { $D1_{2,1} \equiv true : X_2,X_1 / Y_1$, $D2_{2,1} \equiv X_3 : X_5, X_4 / Y_2$,
$D2_{2,2} \equiv X_4 : X_5, X_3 / Y_2$,
$D2_{2,3} \equiv X_5 : X_4, X_3 / Y_2$ }

Level 3: { $D2_{3,1} \equiv true : X_5, X_4, X_3 / Y_2$ }

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

- if $W=\{X_1, X_2\}$ then extension $In(2)= W \cup \{Y_1, Y_2\}$ is computed by making the assumption that X_5, X_4 and X_3 hold ($Out(2)=\{\neg X_5, \neg X_4, \neg X_3\}$) and by applying defaults $D1_{0,1}$ and $D2_{3,1}$ respectively, i.e. $\Pi(2)=\{D1_{0,1}, D2_{3,1}\}$. Note that, the default $D1_{0,1}$ takes priority over the default $D2_{3,1}$, due to the fact that the first one does not employ any assumptions while the second one employs three assumptions in the inference process.
- if $W=\{X_1, X_2, X_3\}$ then extension $In(2)= W \cup \{Y_1, Y_2\}$ is computed by making the assumption that X_5 and X_4 hold ($Out(2)=\{\neg X_5, \neg X_4\}$) and by applying defaults $D1_{0,1}$ and $D2_{2,1}$ respectively, i.e. $\Pi(2)=\{D1_{0,1}, D2_{2,1}\}$. Also, note that, the default $D1_{0,1}$ takes priority over the default $D2_{2,1}$.
- if $W=\{X_1, X_3, X_4, X_5\}$ then extension $In(2)= W \cup \{Y_2, Y_1\}$ is computed by making the assumption that only X_2 holds ($Out(2)=\{\neg X_2\}$) and by applying defaults $D2_{0,1}$ and $D1_{1,1}$ respectively, i.e. $\Pi(2)=\{D2_{0,1}, D1_{1,1}\}$. The default $D2_{0,1}$ takes priority over the default $D1_{1,1}$, due to the fact that the first one does not employ any assumptions while the second one employs an assumption in the inference process.
- if $W=\{X_1, X_3, X_4\}$ then extension $In(2)= W \cup \{Y_1, Y_2\}$ is computed by making the assumptions that X_2 and X_5 hold ($Out(2)=\{\neg X_2, \neg X_5\}$) and by applying defaults $D1_{1,1}$ and $D2_{1,1}$ respectively, i.e. $\Pi(2)=\{D1_{1,1}, D2_{1,1}\}$. Now, note that, defaults $D1_{1,1}$ and $D2_{1,1}$, employ the same number of assumptions in the inferences process. Due to this fact and according to the priority criterion on the basis of the total number of assumptions employed by a rule, none of the rules takes priority over the other. Thus, both process $\Pi(2)=\{D1_{1,1}, D2_{1,1}\}$ and $\Pi(2)=\{D2_{1,1}, D1_{1,1}\}$ are feasible. It just happens in this case that processes have identical final impacts to the environment, i.e. $In(2)= W \cup \{Y_1, Y_2\}$ and $Out(2)=\{\neg X_2, \neg X_5\}$ or $In(2)= W \cup \{Y_2, Y_1\}$ and $Out(2)=\{\neg X_5, \neg X_2\}$.

This last example indicates the need for additional priority criteria. For instance, we may use as a criterion the size of factual knowledge a rule employs, i.e. the number of prerequisites. In this case the default $D2_{1,1}$ takes priority over the default $D1_{1,1}$, due to the fact that the first one fires on a larger factual basis in contrast to the second one, although both of them employ the same number of assumptions in the inference process.

Note that although a level may contain two or more defaults that correspond to the same initial contract rule (e.g. $D2_{1,1}$ or $D2_{1,2}$ or $D2_{1,3}$) there is no need for some kind of prioritization among these defaults. If two or more defaults of the same level, which are derived from the same initial rule (i.e. they belong to the same level within the same single-rule KH structure), were to apply simultaneously, then the more general default contained in the immediately lower level should have applied.

Also, 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 is precisely because an agent 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 prerequisites 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 level at a time in a stepwise manner, and the agent moves upwards to the next level of the multi-rule KH structure only when it has exhausted inference at a given level. This ensures that the agent employs the fewest possible hypotheses. We are able to preserve consistency of entailment, if we employ appropriate variations of DfL such as Constrained Default Logic (CDfT) (Schaub, 1992). A CDfT is a DfT that ensures the joint consistency of all justifications involved in reasoning. A default is applied only if its justifications and consequents are consistent with the background theory, i.e., $In(i) \cup \neg Out(i)$. In this case, the possible world model that the agent infers incrementally is the consistent set $In(i) \cup \neg Out(i)$. This is tantamount to saying that the new possible world models inferred by the agent contain, besides previously available knowledge, both the consequents and the assumptions of the defaults that the agent applied.

Finally, note that the technique described here resembles, in a way, stratification of a DfT (Cholewinski, 1994). 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 prerequisite 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 thouroughly.

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.

3.4. Example

For the sake of generality we have, so far, presented our approach to dynamic assumption identification and deployment in abstract terms. In this section we present a flavour of a real example from the application area of e-commerce. We omit, here, a full representation in EC (or some other temporal logic), in order to avoid distracting the reader with details, but an

interested reader is referred to (Giannikis and Daskalopulu, 2006, Giannikis and Daskalopulu, 2007). 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 for purchasing a certain product. Consequently, the seller agent communicates with a carrier agent (CA) and establishes a separate agreement for the safe and timely delivery of goods to the buyer agent. An extract of the initial set of contract norms for the agreement between the buyer agent and the seller agents is as follows:

R={ R1 ≡ SAIsObligedToDeliverToBAWithinNext20days ← BAOrdersFromSA
∧ E-shopFunctionsWell,

R2 ≡ BAIsObligedToPayCAOnBehalfOfSA ← BAOrdersFromSA ∧ CADeliversToBA
∧ CAIsEmpoweredToAcceptPaymentFromBAOnBehalfOfSA }

Note that these norms have the same number of conditions as the norms considered in the abstract example presented in section 3.3. Thus, the corresponding KH structures are as follows:

Single-rule KH structure for R1:
Level 0: {
$D1_{0,1}$ ≡
BAOrdersFromSA, E-shopFunctionsWell
: true
/ SAIsObligedToDeliverToBAWithinNext20days }

Level 1: {
$D1_{1,1}$ ≡
BAOrdersFromSA
: E-shopFunctionsWell
/ SAIsObligedToDeliverToBAWithinNext20days,

$D1_{1,2}$ ≡
E-shopFunctionsWell
: BAOrdersFromSA
/ SAIsObligedToDeliverToBAWithinNext20days }

Level 2: {
$D1_{2,1}$ ≡
true
: E-shopFunctionsWell, BAOrdersFromSA
/ SAIsObligedToDeliverToBAWithinNext20days }

Single-rule KH structure for R2:
Level 0: {
$D2_{0,1}$ ≡

BAOrdersFromSA, CADeliversToBA, CAIsEmpoweredToAcceptPaymentFromBAOnBehalfOfSA
: true
/ BAIsObligedToPayCAOnBehalfOfSA }

Level 1: {
$D2_{1,1}$ ≡
BAOrdersFromSA, CADeliversToBA
: CAIsEmpoweredToAcceptPaymentFromBAOnBehalfOfSA
/ BAIsObligedToPayCAOnBehalfOfSA,

$D2_{1,2}$ ≡
BAOrdersFromSA, CAIsEmpoweredToAcceptPaymentFromBAOnBehalfOfSA
: CADeliversToBA
/ BAIsObligedToPayCAOnBehalfOfSA,

$D2_{1,3}$ ≡
CADeliversToBA, CAIsEmpoweredToAcceptPaymentFromBAOnBehalfOfSA
: BAOrdersFromSA
/ BAIsObligedToPayCAOnBehalfOfSA }

Level 2: {
$D2_{2,1}$ ≡
BAOrdersFromSA
: CAIsEmpoweredToAcceptPaymentFromBAOnBehalfOfSA, CADeliversToBA
/ BAIsObligedToPayCAOnBehalfOfSA,

$D2_{2,2}$ ≡
CADeliversToBA
: CAIsEmpoweredToAcceptPaymentFromBAOnBehalfOfSA, BAOrdersFromSA
/ BAIsObligedToPayCAOnBehalfOfSA,

$D2_{2,3}$ ≡
CAIsEmpoweredToAcceptPaymentFromBAOnBehalfOfSA
: CADeliversToBA, BAOrdersFromSA
/ BAIsObligedToPayCAOnBehalfOfSA }

Level 3: {
$D2_{3,1}$ ≡
true
: CAIsEmpoweredToAcceptPaymentFromBAOnBehalfOfSA, CADeliversToBA, BAOrdersFromSA
/ BAIsObligedToPayCAOnBehalfOfSA }

Multi-rule KH structure for R1 and R2:
Level 0: {
$D1_{0,1}$ ≡
BAOrdersFromSA, E-shopFunctionsWell

: true
/ SAIsObligedToDeliverToBAWithinNext20days,

$D2_{0,1}$ ≡
BAOrdersFromSA, CADeliversToBA, CAIsEmpoweredToAcceptPaymentFromBAOnBehalfOfSA
: true
/ BAIsObligedToPayCAOnBehalfOfSA }

Level 1: {
$D1_{1,1}$ ≡
BAOrdersFromSA
: E-shopFunctionsWell
/ SAIsObligedToDeliverToBAWithinNext20days,

$D1_{1,2}$ ≡
E-shopFunctionsWell
: BAOrdersFromSA
/ SAIsObligedToDeliverToBAWithinNext20days,

$D2_{1,1}$ ≡
BAOrdersFromSA, CADeliversToBA
: CAIsEmpoweredToAcceptPaymentFromBAOnBehalfOfSA
/ BAIsObligedToPayCAOnBehalfOfSA,

$D2_{1,2}$ ≡
BAOrdersFromSA, CAIsEmpoweredToAcceptPaymentFromBAOnBehalfOfSA
: CADeliversToBA
/ BAIsObligedToPayCAOnBehalfOfSA,

$D2_{1,3}$ ≡
CADeliversToBA, CAIsEmpoweredToAcceptPaymentFromBAOnBehalfOfSA
: BAOrdersFromSA
/ BAIsObligedToPayCAOnBehalfOfSA }

Level 2: {
$D1_{2,1}$ ≡
true
: E-shopFunctionsWell, BAOrdersFromSA
/ SAIsObligedToDeliverToBAWithinNext20days,

$D2_{2,1}$ ≡
BAOrdersFromSA
: CAIsEmpoweredToAcceptPaymentFromBAOnBehalfOfSA, CADeliversToBA
/ BAIsObligedToPayCAOnBehalfOfSA,

$D2_{2,2}$ ≡
CADeliversToBA
: CAIsEmpoweredToAcceptPaymentFromBAOnBehalfOfSA, BAOrdersFromSA
/ BAIsObligedToPayCAOnBehalfOfSA,

$D2_{2,3}$ ≡
CAIsEmpoweredToAcceptPaymentFromBAOnBehalfOfSA
: CADeliversToBA, BAOrdersFromSA
/ BAIsObligedToPayCAOnBehalfOfSA }

Level 3: {
$D2_{3,1}$ ≡
true
: CAIsEmpoweredToAcceptPaymentFromBAOnBehalfOfSA, CADeliversToBA, BAOrdersFromSA
/ BAIsObligedToPayCAOnBehalfOfSA }

Suppose that the current explicit knowledge that the buyer agent possesses is that it has ordered goods from the seller agent, that the e-shop functions properly, and that the carrier agent that will actually deliver the goods is legally empowered to accept payment on behalf of the seller agent, i.e., the buyer agent's current knowledge is:

W={ BAOrdersFromSA, E-shopFunctionsWell,
CAIsEmpoweredToAcceptPaymentFromBAOnBehalfOfSA }

On the basis of this knowledge alone, the buyer may only infer, that the seller is obliged to deliver products to it, within the next 20 days, i.e. the extension In(1)= W ∪ { SAIsObligedToDeliverToBAWithinNext20days } is computed by making no assumptions (Out(1)={ }) and by applying default $D1_{0,1}$, i.e. Π(1)={ $D1_{0,1}$ }.
But, apart from establishing what it must expect from its counterparty, the buyer agent may wish to explore potential future scenaria. For instance, the buyer may need to perform best-guess reasoning and plan its future activities on the assumption that certain events/actions will occur, and that its partners' actions will be valid. Suppose that the buyer wants to infer the time by which it will have to pay for the goods, assuming that all goes well and it receives them in good time, because it wants to plan to have adequate funds available. To derive such an answer the buyer agent needs to identify and employ the assumption that delivery happens in due time (CADeliversToBA)[2], i.e. the extension In(2)= W ∪ { SAIsObligedToDeliverToBAWithinNext20days, BAIsObligedToPayCAOnBehalfOfSA } is computed by making the assumption that CADeliversToBA holds (Out(2)={ ¬CADeliversToBA }) and by applying defaults $D1_{0,1}$ and $D2_{1,2}$ (Π(2)={ $D1_{0,1}$, $D2_{1,2}$ }), respectively.
Now suppose that the buyer agent does not possess complete historical information, i.e. it does not know everything that may have happened so far. Let its current knowledge be such

[2] In the full representation of the example, using some temporal logic, the temporal conditions involved in norms, are treated as all other conditions, when the agent constructs single-norm KH structures, i.e. the agent can make assumptions about them as well.

that it only knows that it ordered goods from the seller agent, that the e-shop functions well, and that the carrier agent delivered goods to it.

W={ BAOrdersFromSA, E-shopFunctionsWell, CADeliversToBA }

The buyer may need to perform no-risk reasoning, in order to derive a conclusion based on assumptions, because alternatively it might find itself in an undesirable situation. For instance, it may want to infer that it has an obligation to pay for the goods that it received, yet this inference is not possible, unless it assumes that the carrier agent is legally empowered to accept payment on behalf of the seller agent (CAIsEmpoweredToAcceptPaymentFromBAOnBehalfOfSA), i.e. the extension In(2)= W ∪ { SAIsObligedToDeliverToBAWithinNext20days, BAIsObligedToPayCAOnBehalfOfSA } is computed by making the assumption that CAIsEmpoweredToAcceptPayment FromBAOnBehalfOfSA holds (Out(2)={¬CAIsEmpoweredToAcceptPaymenFromBAOn BehalfOfSA}) and by applying defaults $D1_{0,1}$ and $D2_{1,1}$ (Π(2)={ $D1_{0,1}$, $D2_{1,1}$ }), respectively. In this scenario, the buyer agent does not possess knowledge about the carrier agent's legal power to accept payment on behalf of the seller agent. It may be the case that when such information was communicated to it by the seller agent, it got lost or distorted, or it may be the case that the seller agent simply 'forgot' to communicate such information to it. If the buyer agent does not perform no-risk reasoning, it risks finding itself in a situation where it will have violated its obligation to pay for the goods that it received, inadvertently, and it will have to face the legal consequences, e.g. to pay extra charges.

4. Related Work on Assumption-based Reasoning

During the past thirty years or so various approaches to assumption-based reasoning have been proposed in the Artificial Intelligence literature. These can be broadly grouped into:

- those that rely on *a priori specification* of the assumptions that can be employed during the reasoning process, i.e., those where assumption identification is static; and
- those that attempt to support *ad hoc identification* of potentially useful assumptions during the reasoning process, that is those that purport to identify and employ assumptions dynamically.

Our approach, which is presented in section 3, is clearly related closely to the second group. However, we review here static approaches as well, since they form the basis on which dynamic approaches to assumption-based reasoning were developed. In order to assist readers to familiarize themselves both with the motivations for assumption-based reasoning and with the technical aspects of the various approaches, we found it useful to include static approaches in our discussion.

4.1 Static Assumption-based Reasoning

Doyle in 1979 (Doyle, 1979) described the representation and structure of a Truth Maintenance System (TMS). He argued that his work solves part of the belief revision problem and provides a mechanism for making assumptions. It is guided by the so called problem of control, that is the problem of deciding on what the system's next inference will be. 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 TMS, 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 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 (de Kleer, 1986a, de Kleer, 1986b) presented a new kind of TMS that avoids certain previous pitfalls. Contrary to (Doyle, 1979) this new approach, the Assumption-based Truth Maintenance System (ATMS), is based on manipulating not only justifications but assumptions as well. In this way, each belief is labelled with the set of assumptions under which it holds, besides the justifications that support it. Later, Reiter and de Kleer, in (Reiter and de Kleer, 1987) and (de Kleer, 1988) respectively, proposed some extensions and generalizations of the ATMS that are concerned mainly with the way the system is able to manipulate clauses, which are more general than Horn clauses. Based on the above ideas of TMS and ATMS, Kohals *et al.* in (Kohlas and Monney, 1993, Anrig *et al.*, 1997) 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 (Poole *et al.*, 1987, Poole, 1988) 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 G is explainable from F and Δ, if there is a set D of ground instances of Δ such that $F \cup D$ entails G, and $F \cup D$ is consistent[3]. Finally, in (Poole, 1996) 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 that the agent faces, i.e. planning, diagnosis or default reasoning. Although, this approach is close to the technique that is presented in this chapter, there is a quite important difference. In Theorist, predefined rules determine what can be used as hypotheses, while in ours an agent discovers candidate hypotheses for itself.

Bondarenko *et al*. in (Bondarenko *et al.*, 1993) proposed an argumentation-based approach to hypothetical reasoning. This work is inspired by Dung's general argumentation framework and it is based on the notions of attack and counterattack of 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.

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

Kowalski and Sadri in (Kowalski and Sadri, 1994, Kowalski and Sadri, 1997) compare the Situation Calculus (McCarthy, 1963, Reiter, 1993) and the EC. Both calculi are formulated as Logic Programs. As noted, the EC was intended primarily for reasoning about actual events, and the Situation Calculus 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. When events are asserted in the knowledge base it is important to verify its integrity, and to this end integrity constraints are used to ensure that i) an event that happens is possible given the current situation, that is all its associated preconditions actually or hypothetically hold; and ii) no concurrent events are possible. These constraints play different roles in the case of actual or hypothetical events. In the first case, they ensure that only possible events happen, and, in the second case, they define the context in which an assumption is possible.
Provetti in (Provetti, 1996) also deals with the problem of actual and hypothetical actions in terms of the Situation Calculus and the EC and introduces new predicates such as HypHolds(fluent,situation) to denote that a fluent is assumed to hold in a situation, as well as new ordered types of constants. 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 (Florea, 1997) presents an assumption-based reasoning approach for multi-agent systems that is based on the TLI (Teoria Logica Implicita) logic. The proposed logic is first-order logic enhanced with special notation for the representation of Reiter's original default rules and for the derivation of extensions.
Tahara in (Tahara, 2004) addresses the issue of inconsistency that may arises in the knowledge base as a result of inconsistent hypotheses and uses a preference ordering in order to resolve contradictions.

4.2 Dynamic Assumption-based Reasoning

The most notable approaches that fall into the second category, where it is attempted to identify and employ assumptions dynamically, include those of Cox and Pietrzykowski (Cox and Pietrzykowski, 1986), Reichgelt and Shadbolt (Reichgelt and Shadbolt, 1989, Reichgelt and Shadbolt, 1990), Abe (Abe, 1999), Pellier and Fiorino (Pellier and Fiorino, 2004, Pellier and Fiorino, 2005) and Jago (Jago, 2005). Our work is, obviously, related mostly to this second category. However, it seems to us that assumption identification in these approaches is not truly dynamic. Before we discuss briefly each of these approaches, we make some general remarks on this issue: Some of these approaches rely on the use of a pre-specified pool of assumptions, from which the agent must choose appropriate ones, whenever it identifies an information gap and needs to fill it, in order to proceed with its reasoning. A natural question that arises though, is whether it is realistic to expect that candidate assumptions can be identified in advance. It may be the case that in some application domains this is possible. However, in such cases, candidate assumption identification is not really dynamic, rather selection of an appropriate assumption from the pre-specified pool, may be carried out dynamically during the inference process. This selection though, requires deductive proof, which is notably computationally expensive.

Other dynamic approaches that purport to support dynamic identification of assumptions, rely on finding appropriate assumptions in a goal-driven manner, that is, a particular conclusion that the agent wants to derive is given, and then the agent identifies the assumptions that are required, in order for this conclusion to be derivable. In some cases, such goal-driven identification of candidate assumptions requires proof. But more importantly, the problem that we perceive with purely goal-driven assumption identification is the following: although software agents, in general, are inherently goal-driven in planning their activity, their rationality (and consequently their performance measure) depends on the extent to which they are perceptive of their environment, so that they may exploit changes in it. A purely goal-driven identification of candidate assumptions does not leave much room for the agent to adapt to circumstances.

We now discuss each one of the approaches on dynamic assumption identification and usage, with some additional comments on each of them:

Cox and Pietrzykowski in (Cox and Pietrzykowski, 1986) explore the problem of the derivation of hypotheses to explain observed events. This is equivalent to finding what assumptions together with some axioms imply a given formula. This is similar to what we refer to as no-risk reasoning, i.e. the identification and usage of assumptions about the past. In this work, the identification of assumptions is essentially goal-driven, and it requires proof, in order to establish that the observed event is implied by what is known (the axioms) and what is assumed.

Reichgelt and Shadbolt in (Reichgelt and Shadbolt, 1989, Reichgelt and Shadbolt, 1990) present a way to analyze planning as a form of theory extension. Theory extension enables an agent to add further assumptions to its knowledge base, in order to derive potential plans towards goal achievement. This is similar to what we refer to as best-guess reasoning, i.e. the identification and usage of assumptions about the future. Their approach requires the use of a pre-specified assumption pool, where candidate assumptions are defined in advance, along with preconditions for their usage. The selection of an appropriate assumption from this pool is conducted in a goal-driven manner and requires that the preconditions associated with the assumption may be deductively proved from the knowledge base. If multiple assumptions have preconditions that are satisfied, selection amongst them is performed by checking them against pre-specified criteria, e.g. parsimony (the assumption with the fewest preconditions is selected) or generality (the more general assumption is preferred).

Abe in (Abe, 1999), also, deals with the problem of missing hypotheses for the explanation of an observation. He proposes a way to generate analogous hypotheses from the knowledge base when the latter lacks the necessary ones. This work extends the Clause Management System (CMS) proposed by Reiter and de Kleer (Reiter and de Kleer, 1987) for abduction. A CMS, given an observation O that cannot be explained from the knowledge base KB ($KB \nvDash O$), returns as set of minimal clauses O′ such that $KB \vDash O \cup O'$ and $KB \nvDash O'$. 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. Hypothesis generation is done in two distinct steps: i) using first abduction and then deduction, candidate hypotheses are searched in the knowledge base, and ii) in case where such candidate assumptions do not exist in the knowledge base, analogous hypotheses are generated by examining clauses in the knowledge base and the assumption requirements that were identified in the previous step. Hypotheses are generated ad hoc during the inference process, by exploiting predefined analogy

relationships between clauses. This is an attractive approach, but it requires caution: in some applications it is difficult to define analogy relations between clauses, in advance; if no such definition for analogy is provided a priori, counterintuitive results may be produced: For instance, suppose that a buyer agent is obliged to pay a seller agent by some deadline, and that it actually proceeds to do so by cash deposit into the seller's bank account. Although the action of paying via a cash deposit is analogous to the action of paying in cash (in the sense that they have the same practical effect, the seller agent ends up possessing the required funds), the contract that regulates the exchange between the two agents may dictate that only payment in some specific form is deemed as acceptable. The two distinct forms of payment that seem analogous in terms of practical effects, may have different legal effects: one will result in the successful discharge of the buyer's obligation to pay the seller, while the other will result in a (technical) violation of this obligation.

Pellier and Fiorino in (Pellier and Fiorino, 2004, Pellier and Fiorino, 2005) address Assumption-based Planning, and propose a mechanism by which an agent can produce "reasonable" conjectures, i.e. assumptions, based on its current knowledge. Any action precondition that cannot be proved from the knowledge base is considered to be a candidate assumption. A tentative plan (i.e. one that involves assumptions) becomes firm, and can be employed by the agent in order to achieve a specific goal, only when the agent can satisfy all of the conjectures, and this requires the agent to regard them as sub-goals and produce plans for them in turn.

Jago in (Jago, 2005) uses the notion of context in making assumptions. A context is the current set of the agent's beliefs. Nested contexts are used to model nested assumptions, and temporally ordered contexts are used to represent the agent's set of beliefs as it changes over time. Assumptions are not identified a priori, but rather during the reasoning process, either by guessing or in a goal-driven manner.

5. Conclusions

The work presented in this chapter is motivated by the need for assumption-based reasoning in open normative multi-agent environments. The behaviour of agents in multi-agent environments is restricted by the norms that regulate the particular environment in which they participate. In the most general case, regardless of any particular application domain, some communication and interaction protocols govern the society of agents; specific application domains may require additional prescription of agent behaviour, and pose application-specific norms. Unavoidably in open environments agents have incomplete knowledge about their world, and about other agents, yet they must somehow plan their activities (both private and public), and they must somehow preserve their autonomy, i.e. decide for themselves which behaviour serves their private or shared goals in the best way. We believe that the degree of agent autonomy is related to the extent to which an agent is 'free' to make assumptions about anything it does not know about, and we want to support assumption identification and usage, without a priori restrictions on the agent, and without resorting to proof, which is prohibitive computationally.

We have developed a prototype implementation, in order to establish that our proposal is feasible. One natural direction for future work is the extension of our prototype to handle variables and their quantification, and we are currently investigating four major approaches (cf. (Reiter, 1980, Lifschitz, 1990, Poole, 1988, Kaminski, 1995, Kaminski *et al.*, 1998)) to the

semantics of open Default Theories, to establish what might be appropriate for computational purposes.

Another direction for future work is to explore whether our ideas about the dynamic and ad hoc identification and usage of candidate assumptions via the construction of hierarchical multi-level structures, can be applied to other approaches to nonmonotonic reasoning such as Logic Programs (Gelfond and Lifschitz, 1988, Gelfond and Lifschitz, 1991) and Defeasible Logic (Nute, 1994).

Finally, we have already started exploring alternative ways for representing the possible knowledge/hypothesis states of an agent as lattices, which can be traversed both upwards and downwards, reflecting an agent's expanding or contracting knowledge base, or equivalently an agent's contracting or expanding assumption requirements; we are experimenting with the computational implementation of the associated algorithms for such traversal and have recorded some preliminary results in (Giannikis and Daskalopulu, 2009).

6. References

Abe, 1999.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.

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

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

Artikis *et al.*, 2002. Alexander Artikis, Jeremy Pitt, and Marek J. Sergot. Animated specifications of computational societies. In *1st International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1053–1061, Bologna, Italy, July 15-19 2002. ACM.

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

Cholewinski, 1994. 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.

Cox and Pietrzykowski, 1986. Philip T. Cox and Tomasz Pietrzykowski. Causes for events: Their computation and applications. In Jǫrg 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.

de Kleer, 1986a. Johan de Kleer. An assumption-based TMS. *Artif. Intell.*, 28(2):127–162, 1986.

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

de Kleer, 1988. 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.

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

Farrell *et al.*, 2005. Andrew D. H. Farrell, Marek J. Sergot, Mathias Sallı, and Claudio Bartolini. Using the event calculus for tracking the normative state of contracts. *Int. J. Cooperative Inf. Syst.*, 14(2-3):99–129, 2005.

Florea, 1997. 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.

Gelfond and Lifschitz, 1988. Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. pages 1070–1080, 1988.

Gelfond and Lifschitz, 1991. Michael Gelfond and Vladimir Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Comput.*, 9(3/4):365–386, 1991.

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

Giannikis and Daskalopulu, 2007. Georgios K. Giannikis and Aspassia Daskalopulu. The representation of e-Contracts as default theories. In H.G. Okuno and M. Ali, editors, *Proceedings of 19th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (IEA/AIE 2007)*, LNAI 4570, pages 963–973, Kyoto, Japan, 2007. Springer-Verlag Berlin Heidelberg.

Giannikis and Daskalopulu, 2008. Georgios K. Giannikis and Aspassia Daskalopulu. How agents know what to assume when? In *IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2008)*, pages 538–545, Sydney, Australia, 9 - 12 December 2008. IEEE Computer Society.

Giannikis and Daskalopulu, 2009. Georgios K. Giannikis and Aspassia Daskalopulu. E-contracting agents reasoning hypothetically and nonmonotonically. Technical Report 7122/1, A006.3ΓΙΑ, Department of Computer and Communications Engineering, University of Thessaly, 7122/1, A006.3ΓΙΑ, April 2009.

Hewitt, 1985. Carl Hewitt. The challenge of open systems: current logic programming methods may be insufficient for developing the intelligent systems of the future. *BYTE*, 10(4):223–242, 1985.

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

Jean-Yves Girard and Lafont, 1989. Paul Taylor Jean-Yves Girard and Yves Lafont. *Proofs and types*. Cambridge University Press, New York, NY, USA, 1989.

Kaminski *et al.*, 1998. 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.

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

Kohlas and Monney, 1993. Jȯrg Kohlas and Paul-Andrı Monney. Probabilistic assumption-based reasoning. In David Heckerman and E. H. Mamdani, editors, *UAI*, pages 485–491. Morgan Kaufmann, 1993.

Kowalski and Sadri, 1994. 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.

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

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

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

Marvn and Sartor, 1999. Rafaèl Hernández Marín and Giovanni Sartor. Time and norms: a formalisation in the event-calculus. In *7th International Conference on Artificial Intelligence and Law*, pages 90–99, New York, NY, USA, 1999. ACM Press.

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

McCarthy, 1980. John McCarthy. Circumscription - a form of non-monotonic reasoning. *Artif. Intell.*, 13(1-2):27–39, 1980.

Nute, 1994. Donald Nute. Defeasible logic. In Dov Gabbay, Christopher J. Hogger, and J. A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming, Nonmonotonic Reasoning and Uncertain Reasoning*, volume 3, pages 353–395. Oxford University Press, 1994.

Pellier and Fiorino, 2004. 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.

Pellier and Fiorino, 2005. 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.

Poole *et al.*, 1987. 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.

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

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

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

Reichgelt and Shadbolt, 1989. Han Reichgelt and Nigel Shadbolt. Planning as theory extension. In *Proceedings of the Seventh Conference on Artificial Intelligence and Simulation of Behaviour (AISB89)*, pages 191–199, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc.

Reichgelt and Shadbolt, 1990. Han Reichgelt and Nigel Shadbolt. A specification tool for planning systems. In *European Conference on Artificial Intelligence (ECAI90)*, pages 541–546, 1990.

Reiter and de Kleer, 1987. 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.

Reiter, 1977. Raymond Reiter. On closed world data bases. In *Logic and Data Bases*, pages 55–76, 1977.

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

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

Rouached *et al.*, 2005. Mohsen Rouached, Olivier Perrin, and Claude Godart. A contract-based approach for monitoring collaborative web services using commitments in the event calculus. In Anne H. H. Ngu, Masaru Kitsuregawa, Erich J. Neuhold, Jen-Yao Chung, and Quan Z. Sheng, editors, *6th International Conference on Web Information Systems Engineering*, volume 3806 of *Lecture Notes in Computer Science*, pages 426–434. Springer, 2005.

Schaub, 1992. Torsten Schaub. On constrained default theories. In *17th European Conference on Artificial Intelligence (ECAI'92)*, pages 304–308, 1992.

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

Web Intelligence and Intelligent Agents

Edited by Zeeshan-Ul-Hassan Usmani

ISBN 978-953-7619-85-5
Hard cover, 486 pages
Publisher InTech
Published online 01, March, 2010
Published in print edition March, 2010

This book presents a unique and diversified collection of research work ranging from controlling the activities in virtual world to optimization of productivity in games, from collaborative recommendations to populate an open computational environment with autonomous hypothetical reasoning, and from dynamic health portal to measuring information quality, correctness, and readability from the web.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen