Improving Assumption based Distributed Belief Revision*

Benedita Malheiro† and Eugénio Oliveira‡
†Dept. of Electrical Engineering, Instituto Superior de Engenharia do Porto,
Rua de S. Tomé, 4200 Porto, Portugal,
Email: mbnm@fe.up.pt
‡Dept. of Electrical Engineering and Computers, Faculty of Engineering,
University of Porto, Rua dos Bragas - 4099 Porto CODEX, Portugal,
Email: eco@fe.up.pt

Abstract. Belief revision is a critical issue in real world DAI applications. A Multi-Agent System not only has to cope with the intrinsic incompleteness and the constant change of the available knowledge (as in the case of its stand alone counterparts), but also has to deal with possible conflicts between the agents' perspectives. Each semi-autonomous agent, designed as a combination of a problem solver – assumption based truth maintenance system (ATMS), was enriched with improved capabilities: a distributed context management facility allowing the user to dynamically focus on the more pertinent contexts, and a distributed belief revision algorithm with two levels of consistency. This work contributions include: (i) a concise representation of the shared external facts; (ii) a simple and innovative methodology to achieve distributed context management; and (iii) a reduced inter-agent data exchange format. The different levels of consistency adopted were based on the relevance of the data under consideration: higher relevance data (detected inconsistencies) was granted global consistency while less relevant data (system facts) was assigned local consistency. These abilities are fully supported by the ATMS standard functionalities.

1 Introduction

Since Aristotle, it has been known that man does not reason with absolute knowledge, but with beliefs [1]. The intrinsic incomplete and dynamic nature of human knowledge forces man to revise his beliefs in face of new discoveries. Intelligent systems that portrait such anthropomorphic behaviour are called *belief revision systems* or *reason maintenance systems*. These problem solving entities are designed to make decisions based on partial, imprecise, and ever changing information. However, in systems in which several agents cooperate with one another within a decentralised control regime (Multi-Agent Systems), the information management problem is exacerbated still further - each agent has to contend with deficiencies and changes in the information supplied by its contemporaries as well as with its own local information.

To keep track of an agent's changing beliefs, researchers have devised a collection of techniques called Truth Maintenance [6]. Truth Maintenance Systems (TMS) main features are

* *In SCAI'95 - 5th Scandinavian Conference on Artificial Intelligence, Frontiers in Artificial Intelligence and Applications, Eds. A. Aamodt and J. Komorowski, IOS Press, 1995.*

the maintenance of the consistency between their beliefs, the reasons for their beliefs, and the identification of contradictions. Whilst these systems are generally sufficient for maintaining beliefs in an asocial context, they need to be extended if they are to be used in a social context. Apart from beliefs that an individual agent has generated for itself, there will be beliefs that it has been informed about by other community members (either because an acquaintance has answered a query or because it has volunteered a piece of relevant information). In such cases a number of crucial decisions must be made about how the information provided by other agents should be treated - should it be given the same credibility as locally deduced beliefs?, should it only be used when there is supporting local evidence?, how should contradictions between the beliefs of different agents be dealt with?, and so on.

The discussion of these issues is presented in the following sections: Section 2 introduces the reader into the pertinent belief revision aspects in a monolithical framework, in Section 3 discusses the extension of the framework to the distributed setting, Section 4 presents the adopted solutions for the developed distributed belief revision system. Finally, in Section 5 the conclusions are drawn.

2 Asocial Belief Revision

In a TMS belief means justified belief – either it is an assumption or it has been deduced from other beliefs. Depending on the scheme chosen for registering the dependencies between beliefs there are single and multiple context TMS: in justification based TMSs (JTMSs) each belief is associated with the beliefs that immediately caused it [4]; whereas in an assumption based TMS (ATMS) each belief is associated with the smallest set of environments from which it can be deduced (*the belief's label*) [1][2]. The selection of the most adequate TMS is problem domain depend. In the case of the work described in this paper, since different agents may hold distinct perspectives over the same issues, the ATMS multiple contexts management facility is preferred.

A belief revision system is composed of two main units, that work under a master/slave relationship, (i) the problem solver; and (ii) the truth maintenance system, respectively. The TMS guarantees that the conclusions reached by the problem solver are always kept updated and consistent. However it only deals with propositions (usually substituted by arbitrary identifiers called nodes) and their dependencies. For each proposition there will be a node and for each dependency a justification which describes how the node was deduced from other nodes [7].

2.1 An Assumption based Belief Revision System

The problem solver inference activity commands the ATMS. The basic operations are the following: (i) creation of a new assumption node whenever a new proposition is assumed to be true; (ii) creation of a new ordinary node when a new proposition is deduced; and (iii) addition of a new justification to an existing node whenever a new way of deducing the node is found. The set of propositions the problem solver dynamically assumes to be true are the system's assumptions. Until there is no evidence that these hypotheses are false they will remain believed. The belief revision system builds its dependency network based on these hypotheses

[1] The whole ATMS operation is based on the dynamic set of propositions externally believed to be true, called assumptions or hypothesis. These propositions will remain assumptions as long as no factual data concerning them is provided. Each deduced proposition has its own label composed of minimal set of environments where it holds. A belief's environment or context is a set of assumptions from which it can be deduced.

and on the existing set of justifications. Each justification defines a dependency relation between the set of antecedent nodes and the consequent node. Formally, justifications are of the type $a_1 \wedge a_2 \wedge ... \wedge a_n \rightarrow c$ where $a_1, ..., a_n$ are the antecedent nodes and c is the consequent node. Such dependencies are only included in the system's dependency network when each antecedent node is believed, or, in other words, when the justification is valid. The responsibility of this continuous inspection and subsequent dispatch of the valid justifications to the ATMS, belongs to the problem solver. Upon the reception of a justification, the ATMS invokes the label updating algorithm and a new environment composed of the union of the antecedent nodes' labels is computed. Then, recursively, every dependent node (including the direct consequent node) updates its label by adding the new environment. The computed environment is the minimal set of assumptions from which, according to the provided justification, the node can be deduced. Once every dependent node has updated its label, each existing ordinary node will own a belief status:

- *unbelieved*, if the label is empty, meaning that there are currently no reasons for believing in the proposition;
- *believed*, if the label is not empty, meaning that there is a least one consistent environment where the proposition holds.

Generally, these belief revision systems are data-driven. This operating mode necessitates the adoption of adequate problem solving methodologies and appropriate knowledge representation. The rules are ordered by dependence. A rule with dependence D is only triggered after the D-1 dependence rules have been triggered, and so forth. The problem solver guarantees that the nodes are generated according to the number of assumptions they depend on, starting with the nodes with the fewest assumptions, based on this dependence information. A successfully instanciated rule is called a consumer. A consumer is triggered only when the ATMS holds valid justifications for each of its precondition nodes; once fired, it converts itself into a justification of the conclusion node and ceases to exist as a consumer. The set of consumers to be executed at any given time can be obtained through a scheduling algorithm [3] and an agenda which contains every node with a non empty label and pending consumers. The problem solver repeatedly chooses one of these consumers, executes it, and then removes it, until there are no nodes left on the agenda.

Simultaneous to the described reasoning mechanism, a continuous consistency checking activity is being undertaken. This is undoubtedly the most important role of any TMS. The problem solver has a body of knowledge entirely dedicated to inconsistency detection – the contradiction detection rules. Every time the problem solver moves into a new point in the search space, the contradiction detection mechanism is automatically triggered to guarantee the consistency of every context under consideration. Whenever a contradiction is spotted by the problem solver, it is immediately communicated to the ATMS where it is marked as an inconsistent set of assumptions, and removed from every nodes' label.

In summary, the activity of an assumption based belief revision system can be viewed as a continuous loop of choosing the consumer with the smallest consistent environment, running it, and then removing it. With such an agenda, the ATMS- problem solver combination will always find the most simple labels first - identifying all possible solutions with the least effort.

2.2 Context Management in an Assumption based Belief Revision System

In real world applications, due to the complexity and to the inherent incompleteness of the information at hand, a judicious use of the available data is essential. To work under these difficult real world circumstances, applications like assumption based belief revision systems have adopted a data driven behaviour - they are characterised by the ability of providing answers or solutions to requests by making use of its currently, yet, presumably insufficient available data resources. On one hand, whenever new facts arrive, the necessary belief revision is triggered and further search space exploration is enabled. On the other hand, in the absence of new data, reasonable guesses are made and additional exploration of the search space is performed.

This operating mode presents a serious problem: the continuing exploration of the search space will, ultimately, lead to combinatory explosion. Even in prior stages one can easily observe the costs of such a policy: how many of the explored contexts are relevant to the current system's activity? how to improve the system's time response to critical requests? How to perform what/if analysis without burdening the system with irrelevant inferences? We propose a new context management mechanism that provides the problem solver with a focusing control over the more relevant contexts in order to partially solve the above mentioned questions.

The system's domain knowledge was divided into sub-domains of expertise. Each sub-domain corresponds to specific sets of rules. This mapping between the sub-domains and the sets of rule is static and is done by the system's developer. Dependencies between the rules of different sub-domains can occur.

These rule groups can be dynamically enabled or inhibited according to the user's wish. We adopted an innovative approach where the rules are represented as nodes with the associated belief status corresponding to their activation status:

- *enabled rules* are believed – represented as assumption nodes;
- *inhibited rules* are unbelieved – represented as ordinary nodes with empty labels.

We not only ensure the adequate focus but we do it in a neat way since this behaviour is exclusively supported by the standard ATMS functionalities. Using this mechanism the problem solver is able to perform context management within the most appropriate sub-domains, exploring only the relevant contexts and achieving a better time response.

3 Distributed Assumption based Belief Revision

The path from the individual to the distributed framework has to be carefully undertaken. While in a monolithic architecture consistency means logical consistency, in a distributed scenario consistency encompasses both logical and physical consistency [2]. In a social context issues like the chosen control regime (centralised or autonomous), the required level(s) of consistency and the adequate amount of inter-agent communication have to be discussed prior to the implementation.

The selection of the system's architecture and the type of inter-agent communication which is acceptable are closely related with the application domain. There are problem domains where centralised control is desirable, some where autonomy is to be preserved at all costs, others where communication is to be kept to its minimum, and finally, others that depend on broadcasting to achieve better solutions quality [8]. Whenever a centralised control is appropriate it is reasonable to build a global TMS which incorporates all of the system's facts and justifications; whereas in the case of a distributed architecture a pragmatic compromise between the achieved consistency level and the information redundancy among the agents has to be reached.

[2] Physical consistency guarantees that whenever a change of belief status of a shared node occurs within an agent, it is immediately communicated to every acquaintance with whom the node is shared.

These design options gave rise to two fundamental approaches in multi-agent belief revision systems:

- *global consistency*, when two or more agents cannot assign a different belief status to the same fact;
- *local consistency*, when different agents may have different perspectives over the same fact if conveniently justified.

In our multi-agent system, each agent is a semi-autonomous agent with a repository of its own, where it records local propositions and justifications. Only when cooperation occurs do non-local facts have to be represented. In such an environment global consistency is unattainable, unless the system broadcasts every relevant activity to all the pertinent agents. However, if the shared data is classified according to its relevance, a satisfiable compromise can be reached. Highly relevant data should be globally consistent while more standard/common information should remain locally consistent. Based on this criteria, contradictions were granted a higher degree of relevance than the facts' belief status. Therefore we settled for the *global sharing of the inconsistencies* and for *local consistency of the facts' belief status*.

3.1 Beliefs Local Consistency

Depending on the composition of the data exchanged, an agent that receives an external fact may or may not receive its label: if the label is sent, it is possible to guarantee the consistency between the foundations of the external fact and the local facts and assumptions; if no label is sent, it is impossible to cross check the external fact's foundations with the local TMS data. In the first case the agents exhibit *local-and-shared well-foundedness* and *local consistency*, whilst in the second case there is only *local well-foundedness* and *local consistency* [5]. We chose the latter based on the minimum communication costs policy: a reduced inter-agent data exchange format where only the nodes and their associated belief status are communicated. Consequently, the community of agents behaves like a democratic society in which each individual can hold a different opinion once it is locally justified – an agent only accepts to revise its beliefs based on external information when it does not have its own convictions regarding that fact.

3.2 Global Inconsistency Sharing

Inconsistent sets of assumptions are called contradictions or *nogoods*. Every time a contradiction is detected by one agent it is thoroughly inspected to determine the set of agents to whom it may be relevant. A locally detected contradiction is relevant to an external agent when the detected set of inconsistent assumptions has been, or may be used, by the external agent during its inference activity.

To establish to which acquaintances a contradiction is relevant, a compilation of the local agent contradiction detection rules is done and a subsequent cross-checking with the acquaintances model is made. The inconsistent set of assumptions is then sent to every external agent where it may be of relevance. An agent, upon the reception of one of these sets, immediately routes it to its ATMS module, where it is registered and processed as if it had been locally detected.

3.3 *Distributed Context Management*

The need for a search focusing mechanism is even more pertinent in a distributed context. Agents may be lead to explore secondary areas not relevant to the system's current objective. There is a genuine risk that an accumulative distracting effect is spread all over the community with the associated unwanted side effects: efficiency overall loss. The system's overall search activity is bounded by the data exchanged during cooperation and the world inputs. This constant updating of the agents' views of the world is achieved through communication, either from directly connected sensors or from the acquaintances.

Whenever a new fact is presented to a recipient agent, it will be used to locally revise its current beliefs and to further explore the search space. As a result of this internal model refinement, the agent informs the acquaintances about the shared internal beliefs that were updated. In the absence of a context management mechanism the agents can easily become absorbed by the not so pressing tasks while the more demanding ones are postponed. To avoid this possible inadequate behaviour, the described search focus mechanism for an individual agent was adapted for the distributed scenario. The necessary mapping between the sub-domains and the groups of rules is static and is contained in the agents' models. Sub-domain control selection is dynamically performed by the user. By default every sub-domain is enabled. The user may inhibit or enable multiple sub-domains according to his wish. When a sub-domain is inhibited its group of rules is disabled. In other words, every rule that belongs to the referred set of rules becomes unbelieved. Conversely, when a sub-domain is enabled each one of its rules becomes an assumption.

4 The Implemented Distributed Belief Revision System

The implemented multi-agent belief revision system is made of a set of semi-autonomous agents. The agents themselves are divided into two distinct functional units: a domain level system and a cooperation layer. The former is implemented as an assumption based belief revision system and contains, among others, expertise on specific domains, planning and scheduling. The latter is the interface between the agent and the rest of the community and provides the necessary facilities for establishing, maintaining and monitoring cooperation [7]. Finally, an User Interface Agent was designed to provide the user/system interaction.

4.1 *Representation of Beliefs*

Within a given agent the existing beliefs are classified according to their scope and origin. The presented classification is essential to the distributed belief revision activity (see next subsection). The different belief categories are:

- *private beliefs* – beliefs the agent has generated and kept to itself;
- *shared beliefs* – beliefs the agent shares with at least one acquaintance. Shared beliefs are subdivided into:
  - *shared internal* or *endogenous* beliefs – beliefs the agent has deduced by itself and are shared with some acquaintance;
  - *shared external* or *exogenous* beliefs – beliefs the agent has received from an acquaintance.

How should external propositions be included in an agent's local dependency network? Since communicated beliefs rely on the reduced *node – belief status* data exchange format, the adopted local representation for external beliefs depends on the associated belief status:

- *external believed facts* are locally represented as *assumptions*;
- *external unbelieved facts* become local *ordinary nodes with empty labels*.

Posterior external beliefs changes are mapped into this representation in a straight forward way:

- a status change from believed to unbelieved – the existing assumption node that was representing the fact is removed and an ordinary node with an empty label is created to represent the new status;
- a status change from unbelieved to believed – the fact's representative ordinary node with an empty label is removed and a new assumption node becomes its new representation.

Incoming contradictions represent inconsistencies that have been externally detected. These inconsistent sets are immediately routed to the local ATMS, where they are registered and processed as if they had been locally detected.

This innovative representation of the communicated facts allows a simple and concise way of incorporating external knowledge, using, exclusively, the already available ATMS's functionalities.

4.2 Distributed Belief Revision

The adopted distributed belief revision implements local consistency for beliefs, and global inconsistency sharing.

As far as beliefs are concerned locality prevails:

- local beliefs (private and shared internal) are locally revised;
- external beliefs (shared external) are revised by their agent of origin.

The local data, composed of the private and shared internal nodes, is revised as in an asocial context (Section 2). However, to comply with the existing social needs, a number of additional features were built. They aim to provide:

- (a) physical consistency – every belief status change of a shared internal node is immediately communicated to all of the interested recipients;
- (b) global inconsistency sharing – every detected inconsistency is routed to all of the relevant acquaintances;
- (c) world model refinement – data received either from the User Interface Agent or from directly connected sensing devices prevails over local beliefs. Although this may seem a violation of the locality privilege, it is a simple substitution of outdated by updated beliefs ;

- (d) local beliefs consistency – whenever a shared external belief is locally deduced it is immediately substituted by its internal counterpart;
- (e) external data homogeneous interpretation – a shared external fact is believed as long as there is one acquaintance where it remains believed, otherwise it is unbelieved [3].

The belief revision of internal facts is embedded within the agent and is performed by the local TMS. External facts belief updating is accomplished when the agent of origin communicates a new belief status to the recipient agents. In this local consistency context, an agent is only allowed to update a fact's belief status, based on external data, if the fact is external and the data has been sent by its agent of origin. The adopted architecture results in locally responsible agents that provide the community with local consistency.

4.3 Cooperation

The already mentioned social activities are upheld by the inter-agent cooperation. A cooperative interaction is started: (i) when an agent needs assistance; (ii) when an agent is able to supply help; and (iii) when a belief revision of shared knowledge occurs. The first type is regarded as task sharing cooperation, while the remaining are mapped into result sharing cooperation. However, an important issue remains unanswered: how do agents know how and when to cooperate? The functionality related to cooperation is represented as a distinct problem solving layer which sits above the ATMS+problem solver layer. The cooperation layer has the following components [10]: (i) a cooperation module; (ii) a communications module which sends/receives messages between the agents; (iii) a self model which represents information about the underlying domain level system; and (iv) an acquaintances model which represents the relevant information about the other community members with which the agent can be expected to interact.

The domain specific knowledge for cooperation is contained in these models ranges from:

- the full specification of the local agent capabilities – in terms of rules, sub-domains of expertise available, mapping of rules into sub-domains, to
- the complete enumeration of every piece of relevant information regarding its acquaintances – inter-agent sub-domains relations, recipients of inconsistencies to share, recipients of beliefs to share, and so forth. The agents are continuously inspecting the acquaintances model to establish if there is someone interested in any of their recent findings or if there is anyone who may provide them with needed data.

This is how the system ensures that the data exchanged between agents is relevant to the problem solving activity of the recipients. Nevertheless, it is left up to the agents (according to the described belief revision mechanism) how to use the incoming data. Once the community has been launched the agents self and acquaintances models are kept static. This cooperation activity is supported through asynchronous selective communication (direct message passing). The used protocol is based on the speech act theory [9].

[3] Belief and disbelief are different from the boolean true and false attributions: while an agent who has no current reasons for believing in a fact declares it unbelieved, another, may still hold valid reasons for believing it.

5 Conclusions

Our major concern was to build an assumption based distributed belief revision system where a reasonable compromise between the amount of inter-agent communication and the attained level of social cohesion could be reached. To achieve this goal the presented distributed belief revision algorithm was based on the following consistency approaches: (i) the facts' local consistency – an incoming fact will only be assimilated by a recipient agent either when the fact does not exist locally (the fact is unknown) or when the fact exists but has not been deduced locally (the fact is external). The belief status of a fact is only revised by the agent that deduced it; (ii) the global inconsistency sharing – every time a contradiction is detected by one agent it is thoroughly inspected to determine the set of agents to whom it may be relevant. The contradiction is then sent to all of the interested recipients where it is automatically registered on their ATMS modules.

The resultant agents exhibit both altruistic and self-centered behaviours: although the locally deduced facts prevail over externally communicated data, there is still a strong community involvement as far as cooperation and contradiction detection is concerned. Special care was taken to ensure that every one of the discussed contributions was fully support by the standard ATMS functionalities. They include: (i) the concise representation of the shared external beliefs (as assumptions when believed and as ordinary nodes with empty labels when unbelieved); (ii) the reduced inter-agent data exchange format based on the communication of the nodes and the associated belief status , and (iii) the simple and effective context management mechanism based on the attribution of belief status to rules. Further improvements are being undertaken: implementation of other consistency approaches and automatic context management. In a near future, we hope to be able to provide a comparative performance analysis based on quantitative results of the implemented features.

References

[1] Armand de Callatay, (1992) "Natural and Artificial Intelligence" North Holland , Elsevier Publishers.

[2] J. de Kleer, (1986) "An Assumption-based TMS" Artificial Intelligence 28 (2), 127- 162.

[3] J. de Kleer, (1986) "Problem Solving with the TMS" Artificial Intelligence 28 (2), 197-224.

[4] J. Doyle, (1979) "A Truth Maintenance System" Artificial Intelligence 12, 231-272.

[5] M. N. Huhns and D. M. Bridgeland, (1991) "Multi-Agent Truth Maintenance" IEEE Trans. on Systems, Man and Cybernetics 21 (6), 1437-1445.

[6] J. P. Martins, (1990) "The Truth, The Whole Truth, and Nothing But The Truth" AI Magazine, Special Issue, 7-25.

[7] B. Malheiro, N. Jennings and E. Oliveira, (1994) "Belief Revision in Multi-Agent Systems", Proc. of the 11th European Conference on Artificial Intelligence, Amsterdam, Holland, 222-249.

[8] C. Mason, (1994) "Introspection as Control in Result-Sharing Assumption-Based Reasoning Agents" in Proc. of the 13th International DAI Workshop, Seattle, USA, 293-317.

[9] J. R. Searle, (1969) "Speech Acts: An Essay in the Philosophy of Language" Cambridge University Press.

[10] T. Wittig (ed.), (1992) "ARCHON: An Architecture for Cooperative Multi-Agent Systems" Ellis Horwood.