The Truth, the Whole Truth, and Nothing But the Truth:  
An Indexed Bibliography to the Literature of Truth Maintenance Systems

João P. Martins

The revision of conclusions drawn during reasoning is an important aspect of intelligent behavior. The study of systems with revisable conclusions has become, in the last ten years, a major area of AI research—non-monotonic reasoning. In 1980 Doyle and London published an indexed bibliography on belief revision. To the best of our knowledge the only similar work published afterwards [Perlis 84], [Perlis 87] deals with non-monotonic reasoning in general. In this paper we try to cover the literature on a subject related both to belief revision and non-monotonic reasoning—truth maintenance systems. It is somewhat difficult to draw a sharp line between papers that are on truth maintenance and papers that just mention it. In collecting the references we tried to stick to the following classes of papers (1) Papers that correspond to overviews, introductions, or tutorials in truth maintenance; (2) Papers that explicitly deal with issues on truth maintenance; and (3) Papers that deal with applications of truth maintenance and discuss the issues at some important length (thus we have excluded papers that reference truth maintenance as a possible technique to be used without any further discussion of how it could be used).

We did not attempt to present the material from a tutorial perspective and thus we assume some previous knowledge in the area of truth maintenance systems. We are also aware that a work of this kind can never be complete and that, if not updated, will become obsolete a short time after publication. We will try to publish updates to the bibliography and we will be very glad to be informed about new work being developed in the area of truth maintenance, as well as work in this area which was not mentioned here due to lack of our knowledge.

Truth maintenance is a collection of techniques for doing belief revision. A truth maintenance system's task is to maintain a set of beliefs in such a way that they are not known to be contradictory and no belief is kept without a reason. Truth maintenance systems were introduced in the late seventies by Jon Doyle and in the last five years there has been an explosion of interest in this kind of systems. In this paper we present an annotated bibliography to the literature of truth maintenance systems, grouping the works referenced according to several classifications.

'I can't believe that!' said Alice  
'Can't you?' the Queen said in a pitying tone.  
'Try again: draw a long breadth, and shut your eyes.'  
Alice laughed. 'There’s no use trying,' she said: 'one can't believe impossible things.'  
'I daresay you haven't had much practice,' said the Queen. 'When I was your age, I always did it for half-an-hour a day. Why, sometimes I've believed as many as six impossible things.'  
Lewis Carroll, Through the Looking Glass
dependency-tracing routines are isolated from the specific problem-solving techniques. Following the work of Doyle several other systems emerged: [McAllester 78, 80] introduced an approach known as LTMS; [Martins 83], [Martins and Shapiro 88] created a system that combines an inference engine with an ATMS-style system; [de Kleer 84, 86a] introduced the ATMS; [Dressler 89] extended the basic ATMS to allow for non-monotonic justifications.

A TMS is usually seen as an independent module that is associated with a problem solver. The problem solver tells the TMS the results of inferences and the TMS’s task is to keep a record of dependencies among propositions and use those dependencies to inform the problem solver in which propositions it should believe. Issues in TMSes concern the study of how to record dependencies among propositions, and, depending on the approach followed we can have three kinds of systems, JTMS, ATMS, and LTMS; Non-monotonicity the study of how to record that the belief on a proposition depends on the disbeliefs on other propositions; Disbelief propagation studies how to disbelieve the consequences of something that is disbelieved (this aspect is usually associated with labeling algorithms); Inference studies how to merge the inference capabilities of a problem solver with the TMS itself—this is somewhat a secondary area of research since traditionally, TMSes are dissociated from the problem solvers; Revision of beliefs studies how to select the culprit for a detected contradiction—again this is another side issue since most of the aspects here fall under the general area of belief revision; Control studies the interaction with a TMS from the problem solver’s perspective; Formal properties of TMSes concerns the study of TMSes as formal systems and their relationship with other non-monotonic reasoning formalisms; and the development of tools that incorporate TMS.

An associated issue that is not covered in this paper, dependency-directed backtracking, can be studied in [Drakos 88] which presents a good overview on dependency-directed backtracking with lots of references.

There is currently considerable interest in truth maintenance systems. This is reflected both by an extensive publication record on truth maintenance systems; by the existence of TMS tutorials at major AI conferences, AAAI-87 [de Kleer, Forbus, and Williams 87], AAAI-88 [McAllester and McDermott 88], IJCAI-89 [de Kleer, Forbus, and McAllester 89], and AAAI-90; and by the existence of specialized workshops, the German Workshop on Truth Maintenance Systems [Stoyan 88], the Reason Maintenance Workshop [Smith and Kelleher 88], and the Workshop on Truth Maintenance Systems held during the European Conference on AI [Martins and Reinfrank 90].

Background work

In this section we list papers that, although do not specifically address TMSes, were influential in their development. Their study helps at understanding the general setting under which TMSes were introduced. Besides the main entries listed in this section, it may be useful to read some of the early papers on the frame problem [McCarthy and Hayes 69], [Hayes 73], and [Raphael 71], as well as the descriptor-indexed bibliography on belief revision [Doyle and London 80].

[de Kleer, Doyle, Steele, and Sussman 77, 79] Describes AMORD an influential system on the ATMS.

[Fikes 75] Presents a successor of STRIPS that uses dependencies among propositions.

[Hayes 75] One of the first descriptions of the need for TMS-style systems.

[London 78] Describes a planning system relying on the use of dependencies. This work was developed at the same time and independently from Doyle’s work.

[Sandewall 67] Presents the first description of an ATMS-like system.

[Stallman and Sussman 77] Describes dependency-directed backtracking and was a direct influence in the development of Doyle’s TMS.

[Winograd 80] A good overview paper that traces the history of process-dependent reasoning in AI systems.

Introductions and Tutorials

In this section we list papers that can be used as a starting point for the study of TMSes.

[Charniak, Riesbeck, and McDermott 80] An introduction to TMS using the JTMS approach. A nice introduction that discusses how to implement a JTMS.

[Charniak, Riesbeck, McDermott, and Meehan 87] An introduction to TMS using the LTMS approach. Discusses the implementation of a LTMS.

[de Kleer and Doyle 82] A small introduction to truth maintenance.

[de Kleer, Forbus, and Williams 87] Tutorial on TMS presented at AAAI 87. Presents complete LISP code for the JTMS, ATMS, and LTMS.
Assumption-based TMS—ATMS

In the systems described in this section the dependencies among propositions (corresponding to beliefs) are recorded by associating each proposition with the non-derived propositions (called hypotheses or assumptions) that underlie its derivation.

[De Kleer 84] The original paper on ATMS. Very well written.
[De Kleer 86a] An in-depth presentation of the ATMS.
[De Kleer 86b] Describes an extension of the basic ATMS to handle non-monotonic justifications, defaults, and disjunctions.
[De Kleer 88] Presents a general labeling algorithm for the ATMS based on negated assumptions.
[Dechter 89] Discusses the complexity of ATMS in a tree-structured domain and identifies the problem’s parameters which have a decisive effect on the complexity.
[Dechter 89b] Discusses the use of guards to control label propagation on the ATMS.
[Doyle 78, 79] The original papers on JTMS.
[Dean and McDermott 87] Describes a system for temporal reasoning in terms of a JTMS.
[Euzenat 89a, 89b, 90] Describes the addition of contexts to the JTMS.
[Junker 89b] Describes a non-monotonic ATMS based on Reiter’s logic that deals with exceptions, inconsistencies, and ambiguity.
[Kelleher 90] Describes how constraint satisfaction problems may be encoded for the ATMS.
[Koff 88], [Koff, Flan, and Dietterich 88] Introduces a specialized ATMS for efficiently computing equivalence relations in multiple contexts.
[Lang 88] Presents a unified approach to ATMS and numerical uncertainty management.
[Martins 83], [Martins and Shapiro 83, 84, 86b,
Articles

Shapiro 88] Describes an ATMS-like system with reasoning capabilities.
[Mason and Johnson 89] Describes DATMS, the distributed ATMS.
[McDermott 83a] Describes a system that integrates JTMS (data dependencies) with ATMS (data pools).
[McDermott 89] Proposes a unifying formalism for JTMS, ATMS, and LTMS.
[Provan 88a, 88c] Presents an extended ATMS incorporating the full Dempster Shafer theory and discusses its application in diagnosis.
[Provan 88b, 88d, 90b] Studies the complexity of the problems that ATMS solves.
[Reiter and de Kleer 87] Presents CMS, Clause Management System, a generalization of ATMS.
[Tayrac 90a, 90b] Describes an ATMS based on CAT resolution.

Logic-based TMS—LTMS
In the systems described in this section propositions (beliefs) are represented as disjunctive clauses of a propositional logic.
[Charniak, Riesbeck, McDermott, and Meehan 87] An introduction to TMS using the LTMS approach. Discusses the implementation of a LTMS.
[McAllester 78, 80] The original papers on LTMS.
[McDermott 89] Proposes a unifying formalism for JTMS, ATMS, and LTMS.

Inference
In this section we list works that attempt at merging the inference capabilities of the problem solver wit the TMS.
[Cravo and Martins 89, 90d] Discusses how to incorporate path-based inference in a ATMS-like system.
[Cravo and Martins 90a, 90b, 90c, 91] Sets up the foundations for the development of a non-monotonic ATMS with reasoning capabilities.
[Junker 88, 89a] Describes the reasoning capabilities of a system that is based on the ATMS.
[Martins 83], [Martins and Shapiro 83, 84, 86b, 88] Describes a system with the capabilities of an ATMS and discusses how it carries out reasoning.
[McAllester 78, 80] The original papers on LTMS. Describe the inference mechanism through the propagation of truth values.
[McAllester 82, 85] Describes RUP, the Reasoning Utility Package, that uses boolean constraint propagation, a technique for extending a partial truth assignment on a proposition.

Non-monotonicity
In this section we list papers that deal with the issue of representing that the belief in one proposition depends on the lack of belief on other propositions.
[Brewka 86] Describes FAULTY, a system capable of non-monotonic reasoning. Briefly compares FAULTY with TMS approaches.
[Cravo and Martins 90a, 90b, 90c, 91] Discusses the use of default rules in an ATMS-like system.
[de Kleer 86b] Describes an extension of the basic ATMS to handle non-monotonic justifications, defaults, and disjunctions.
[Doye 78, 79] The original paper on JTMS. Describe how to handle non-monotonic justifications.
[Dressler 87, 88a, 88b, 89, 90] Discusses an extension of the basic ATMS to handle non-monotonic justifications.
[Junker 89b] Describes a non-monotonic ATMS based on Reiter's default logic.
[Urbanski 88] Discusses an extension of ATMS to incorporate non-monotonic justifications.
[Zetzche 89] Using a certain kind of default rules shows a 1-1 correspondence between the admissible extensions of a JTMS and certain ATMS extensions.

Disbelief propagation
In this section we list papers that deal with the issue of how does one fail to believe all the consequences of a proposition that is disbelieved. There are basically two approaches to this problem: (1) Label-based systems associate a label (typically IN or OUT, respectively, for believed or disbelieved) with each proposition, and disbelief is done by changing the labels; (2) Context-based systems associate contexts with propositions and disbelief is attained by changing the context.
[Borchardt 87] Presents an algorithm for a system similar to a TMS where multiple agents interact.
[Charniak, Riesbeck, and McDermott 80] Presents a labeling algorithm for JTMS, showing that, in the absence of odd loops, it always terminates.
[de Kleer 88] Presents a general labeling algorithm for the ATMS based on negated assumptions.
[Doye 78, 79] The original paper on JTMS. Describes an algorithm for recomputing labels of nodes when a change of belief occurs.
[Dressler and Farquhar 89] Discusses the use of guards to control label propagation on the ATMS.
[Fulcomer and Ball 88, 89b] Presents a labeling method for a parallel TMS. Corrects the algorithm of [Petrie 86b].
[Fulcomer and Ball 89a] Describes a parallel TMS.
Revision of beliefs

In this section we list papers that deal with the issue of picking the culprit for a contradiction.

[Cebulka, Carberry, and Chester 88] Proposes a domain-independent formalism for the selection of the culprit for a contradiction.

[Dhar and Quayle 85] Describes a heuristic procedure to be used in determining what to change in a model when an undesirable condition occurs.

[Doyle 79], [Doyle 89] Discusses general principles for selecting the culprit for a contradiction.

[Doyle 90] Compares theories of belief revision and examines how different limitations on rationality the revision of beliefs.

[Jackson and Pais 90] Presents the Revision-based TMS, a mechanism for computing the minimum revision required of a theory in order to accommodate new information.

[Junker 90] Reviews the backtracking procedure of the TMS to detect sources of odd loop errors and retract assumed labels, allowing the determination of an extension when one exists.

[Nebel 89] A paper on belief revision that tries to link belief revision techniques with TMSes.

[Nebel 90] A book on terminological reason maintenance systems with a good discussion on TMSes.

[Nutter 87] Presents a strategy for restructuring knowledge bases.

[Popchev, Zlatareva, and Mircheva 90] Discusses a special kind of belief revision based on the idea of blocking the propagation of contradictions rather than trying to get rid of them.

[Rao and Foo 89] Discusses theories of belief revision and links them with TMS.

Control

The papers listed in this section deal with the interaction between the TMS and the problem solver, seen from the problem solver's perspective.

[Arlabosse, Jean-Bart, Porte, and Ravinel 88] Presents an architecture for diagnostic problems that uses ATMS.

[Bridgeland and Huhns 90] Describes DTMS, a multi-agent TMS for maintaining logical consistency of beliefs among a group of computational agents that behave non-monotonically.

[Carrol and Tayrac 88] Improves the ATMS by the introduction of a resolution strategy.

[de Kleer and Williams 86a] Describes a backtracking technique to be used with an ATMS.

[Dixon and de Kleer 88, 89] Shows how the combination of a serial machine with a parallel processor can speed up the TMS algorithms.

[Dressler 90] Discusses an architecture for non-monotonic ATMS-based problem solvers.

[Dressler and Farquhar 89] Briefly discusses applications in the diagnosis of analog circuits using guards to control label propagation on the ATMS.

[Dressler and Farquhar 90] Describes an approach that allows the problem solver to maintain a tight control over the contexts explored by the ATMS.

[Forbus and de Kleer 88] Describes a strategy for focusing the search in a system using an ATMS.

[Goodwin 87] Makes a clean distinction between activity (control support) and belief (logical support).

[Koff 88], [Koff, Flan, and Dietterich 88] Introduces a specialized ATMS for efficiently computing equivalence relations in multiple contexts.

[Trum 86] Discusses the control of ATMS.

Formal properties of TMSes

The papers presented in this section are concerned with the study of TMSes as formal systems and their relationship with other non-monotonic reasoning formalisms.

[Brown 85] Describes a logic, PDLD, that characterizes TMSes.

[Brown, Gaucas, and Benanav 87] Introduces lattice-theoretic truth maintenance and show that it subsumes ATMS and JTMS.

[Brown 88] Gives a formal semantics to TMS by offering a logic that characterizes some models of TMSes.

[Brown and Shoham 89] Generalizes and extends the work of [Brown 88].

[Cravo and Martins 90a, 90b, 90c, 91] Describes SWMC, a non-monotonic logic developed to underlie ATMS-like systems.

[Doyle 83a] The first attempt to present a formal description of a TMS. Difficult to read.
[Doyle 83b] An expanded version of [Doyle 83a].

[Elkan 90] Gives a characterization of the inferences performed by a non-monotonic TMS in terms of logic programming with stable set semantics and autoepistemic logic.

[Eshghi 90] Describes a method for computing the stable models of propositional logic programs with negation as failure using the ATMS.

[Freitag and Reinfrank 88] Shows that in certain cases CAPRI, a tool that uses TMS techniques, is sound and complete with respect to a given specification.

[Fujiwara and Honiden 89] Establishes a correspondence between the states acceptable to the TMS and stable expansions of autoepistemic logic.

[Fujiwara and Honiden 90] Describes the semantics of the basic ATMS in terms of propositional Horn Logic.

[Ginsberg 86b] Argues that TMSes correspond to an extension of classical logic with additional truth values.

[Giordano and Martelli 90a] Presents a logic characterization of JTMS in terms of abduction.

[Giordano and Martelli 90c] Presents a logical semantics for JTMS which is able to capture the idea of dependency-directed backtracking.

[Goodwin 84] Presents LCP, Logics of Current proof, designed to be implemented using dependencies and TMSes.


[Haneclou 87] Presents an approach that gives a strict formalization of TMS and is similar to the approach taken in denotational semantics.

[Inoue 90a, 90b] Presents a procedural semantics for the ATMS based on an abductive procedure.

[Inoue 90c] Describes a logical specification of the ATMS and its generalization based on model theory.

[Junker and Konolige 90] Presents a proof procedure for autoepistemic and default logics that uses TMS techniques.

[Kakas and Mancarella 90] Presents a TMS based on abduction.

[Kundu and Chen 89] Characterizes the proper environments for performing in/out labeling in TMS.

[Laskey and Lehner 88, 89] Shows a formal equivalence between Shafer-Dempster theory and ATMS with a probability calculus on the assumptions.

[Levesque 89] Relates abduction with the ATMS.

[Madre and Coudert 90] Presents a TMS whose reasoning is logically complete.

[Martins 83], [Martins and Shapiro 83, 84, 86b, 88] Describes several aspects of an ATMS-like system based on a logic developed to support belief revision systems.

[Martins 90] Presents a revised version of the logic underlying SNeBR, an ATMS-like system.

[Popchev, Zlateva, and Mircheva 89] Presents an abstract-modeling logic adequate to underlie TMSes.

[Popchev, Zlateva, and Mircheva 90] Presents a logical theory of truth maintenance reasoning subsuming plausible reasoning.

[Provan 89] Analyses the theoretical underpinnings for computing Dempster-Shafer belief functions from ATMS labels.

[Provan 90a] Studies (empirically) the computational advantages and disadvantages of JTMS and ATMS techniques and examines the trade-offs of each.

[Provan 90b] Studies the computational complexity of JTMS and ATMS. Presents design choices underlying specific TMSes.

[Reinfrank and Dressler 88], [Reinfrank and Freitag 88] Introduces NMFS (Non Monotonic Formal Systems), a formalism for describing TMS.

[Reinfrank 89a] Discusses the relationship between TMS and non-monotonic logics.

[Reinfrank 89c], [Reinfrank, Dressler, and Brewka 89] Uses NMFS to prove the equivalence between TMS and strongly grounded autoepistemic extensions.

[Sridhar, Murthy, and Krishna 89] Presents an axiomatic system as an underlying formalism for a TMS.

[Witteveen 90a] Describes a partial semantics for TMS and shows how it can be used to generalize the two-valued stable model semantics.

[Witteveen 90b] Presents a characterization of grounded models of a TMS based on fix-point semantics.

Tools that use TMS techniques

There are several commercial tools (mainly for the development of expert systems) that embody some TMS capabilities. This section lists papers describing some of them.

[d’Aloisi, Stock, and Tuozzi 88] Describes KRAPFEN, an hybrid knowledge representation system. Stresses the use of LTMS.

[Clayton 84] Describes ART a tool that provides a multiple context mechanism (called alternative worlds or viewpoints) and a TMS capability.

[Fikes, Nado, Filman, McBride, Morris, Paulson, Treitel, and Yonke 87] Describes the use of an ATMS in KEE.

[Flann, Dietterich, and Corpron 87] Presents FORLOG a logic programming language that uses ATMS.

[Freitag and Reinfrank 87, 88] Discusses an efficient interpreter for CAPRI a tool that uses TMS. Shows that in certain cases CAPRI is sound and complete.
with respect to a given specification.

[Junker 89a] Describes EXCEPT, a toll that is based on the ATMS.

[King, Bigham, Barrett, and Khong 87] Describes MIKIC-TMS, a tool that incorporates LTMS in MIKIC (a general purpose object-oriented inference system).

[Laasri, Maitre, Mondot, Charpillet, and Haton 88] Describes the use of an ATMS (through ART) in hypothetical reasoning.

[McAllester 82] Describes RUP, the Reasoning Utility Package, that uses boolean constraint propagation, a technique for extending a partial truth assignment on a proposition.

[NEURON 86] Describes NEXPERT a tool that keeps dependencies among propositions and performs a kind of truth maintenance.

[Petrie, Russinoff, and Steiner 86] Describes at a shallow level PROTEUS, a tool developed at MCC, that incorporates a TMS.

[Petrie, Russinoff, Steiner, and Ballou 87] Describes PROTEUSc2, a tool developed at MCC, that incorporates a TMS.

[Reinfrank, Beetz, Freitag, and Klug 86], [Reinfrank and Freitag 88] Describes CAPRI, a rule-based non-monotonic inference engine with an integrated TMS.

[Rich 85] Describes CAKE an hybrid reasoning system that uses TMS techniques.

[Vilain 85] Describes KL-TWO a systems that extends McAllester’s RUP.

Applications

In this section we list papers that deal with applications of TMSes.

Reasoning

[Baker and Ginsberg 89] Extends the work of [Ginsberg 88b], [Ginsberg 89] to handle prioritized circumscription.

[Bonte, Castaing, Grandemange, Grumbach, Kayser, and Levy 88] Describes a reasoning system that uses a TMS.

[Etherington 87] Briefly discusses the role of TMSes in formalizing non-monotonic reasoning.

[Friedman 80, 81] Discusses a reasoning system that keeps a record of dependencies as well as crediblity among propositions.

[Ginsberg 86a] Discusses the use of truth maintenance techniques in the computation of counterfactuals.

[Ginsberg 88b, 89] Discusses the application of an ATMS to the construction of a circumscriptive theorem prover.

[Jackson 89] Discusses PABLO, a propositional abductive logic, and compares it with ATMS.

[Laasri, Maitre, Mondot, Charpillet, and Haton 88] Describes the use of an ATMS (through ART) in hypothetical reasoning.

[Martins and Shapiro 86a] Discusses the application of an ATMS-like system to hypothetical reasoning.

[Morris 87a, 88] Discusses Hanks and McDermott’s anomaluous extension problem from the perspective of a TMS.

[Ohta and Inoue 90] Presents an extended production system architecture, based on the ATMS, which can deal with forward reasoning and multiple contexts.

[Reichgelt 88] Proposes an architecture for a reasoning system capable of default reasoning. Its implementation uses TMS techniques.

[Reiter and de Kleer 87] Presents CMS, Clause Management System, a generalization of ATMS and discusses its use in terms of efficiency of search and abductive reasoning.

Constraint Satisfaction

[Bodington and Elleby 88] Discusses how TMSes (both JTMS and ATMS) can be used to solve constraint satisfaction problems.

[Crocker and Dhar 90] Presents a knowledge representation for constraint satisfaction problems and compares it with the TMS approach.

[de Kleer 90] Uses boolean constraint satisfaction to exploit locality in a TMS.

[Dechter and Dechter 89] Discusses distributed algorithms for TMS and their application to constraint satisfaction problems.

[Dhar and Ranganathan 90] Describes the use of TMS in constraint satisfaction problems. Compares this approach with approaches from operations research.

[Inoue and Ohta 90] Describes algorithms for combinatorial search problems involving constraint satisfaction that use TMS techniques.

[Kelleher 90] Describes how constraint satisfaction problems may be encoded for the ATMS.

[Maleki 87] Describes a constraint programming language that uses TMS techniques.

[Mittal and Falkenhainer 90] Describes an ATMS-based implementation of a dynamic constraint satisfaction algorithm.

[Mott, Cunningham, Kelleher, and Gadsen 88] Describes the use of an ATMS for solving a constraint satisfaction problem in scheduling.

[Shanaham and Southwick 89] Discusses applications of TMSes in constraint satisfaction, backtracking, hypothetical reasoning, and theorem proving.

[Smith 88] Discusses the usefulness of ATMS in reducing search in constraint satisfaction problems.

[Stuss 87] Describes a constraint system, based on the ATMS, that can be used for diagnosis or design.
Diagnosis
[Arribas, Jean-Bart, Porte, and Ravinel 88] Describes the use of an ATMS-based architecture in diagnosis.
[Campbell and Shapiro 86] Discusses the use of TMS in the detection of faults in electric circuits.
[de Kleer and Williams 86b] Discusses the use of an ATMS in diagnosis.
[de Kleer and Williams 87] Describes GDE, General Diagnostic Engine, a system for diagnosis based on ATMS.
[d’Ambrosio, Oriati, and Serventi 88] Discusses the use of the TMS capabilities of KEE to the diagnostic process in power plants.
[Dressler and Farquhar 89] Briefly discusses applications in the diagnosis of analog circuits using guards to control label propagation on the ATMS.
[Hamscher 89] Describes a system for troubleshooting of digital circuits that uses an hybrid TMS, resulting from the combination of ideas from ATMS and JTMS.
[Matetz 85] Describes a system for diagnosing multiple faults that uses ATMS techniques (implemented using ART).
[Provan 88a, 88c] Presents an extended ATMS incorporating the full Dempster Shafer theory and discusses its application in diagnosis.
[Puppe 87] Introduces the ITMS (Immediate-Check TMS) and discusses its applications in diagnosis.
[Struss 87] Describes a constraint system, based on the ATMS, that can be used for diagnosis or design.
[Struss 88a] Extends the GDE of [de Kleer and Williams 87] and discusses the use of ATMS in diagnosis.
[Struss 88b] Discusses the use of ATMS in diagnosis.

Planning
[Descotte and Latombe 81] Describes a plan generator for planning the sequence of machine cuts of mechanical parts. Uses TMS techniques.
[Drummond 87] Discusses the use of TMS techniques in planning.
[Fikes, Morris, and Nado 87] Describes a proposal for using TMS techniques in planning.
[Filman 88] Presents an example that solves a planning problem with TMS techniques. The complete description of the example presented in this paper can be found in [Filman 86].
[Janlert 87] Discusses an approach to the frame problem using TMS.
[Kulkarni and Parameswaran 89] Describes the need for TMSes in action representation and modification. Presents a way of representing actions and their effects in a TMS.
[London 78] Describes a planning system relying on the use of dependencies. Describes a system with a functionality similar to a TMS.
[Morris 87b] Presents an approach to the frame problem using a TMS.

Search
[Dechter 88, 90] Describes three techniques that can be used in TMS: backjump, learning while search, and cycle-cutset method.
[Forbus and de Kleer 88] Describes a strategy for focusing the search in a system using an ATMS.
[Inoue 88] Describes a general search algorithm for multiple contexts based on the ATMS.
[Reiter and de Kleer 87] Presents CMS, Clause Management System, a generalization of ATMS and discusses its use in terms of efficiency of search and abductive reasoning.
[Shanaham 87, 88] Describes a Prolog-like theorem prover which records the structure of the search space using data dependencies.
[Shanaham and Southwick 89] Presents an introduction to TMSes and discusses applications in constraint satisfaction, backtracking, hypothetical reasoning, and theorem proving.
[Smith 88] Discusses the usefulness of ATMS in reducing search in constraint satisfaction problems.

User-modeling
[Huang, McCalla, and Greer 90] Presents the Student Model Maintenance System which allows two types of revision in a student model, evolutionary and revolutionary revision.
[Jones and Millington 88] Describes a system that models Unix users using an ATMS.
[Rich 89] Describes the uses of TMS techniques in a system for user modeling.

Miscellaneous
[de Kleer 86c] Discusses how a TMS could be incorporated in a problem solver, presenting an interfacing protocol with the ATMS.
[de Mori and Mong 84] Describes a system for recognizing strings of letters. Uses TMS.
[Ginsberg 88a] Presents an approach based on the ATMS for checking knowledge bases for inconsis-
tency and redundancy.

[Harp and Sederberg 88] Discusses the use of a TMS in configuration problems.

[Hoenkamp 87] Using JTMS, formalizes the phenomenon, observed by psychological experiments, that people often cling to their initial beliefs more strongly than appears warranted.

[Hollan, Hutchins, and Weitzman 84] Describes STEAMER, a computer-based training system that uses TMS techniques.

[Hanj and Venken 87] Describes the use of a TMS-like system in knowledge base management systems.

[Joubel and Raiman 90] Describes an architecture that handles assumptions (in an ATMS-like fashion) and time.

[Kunz, Bonura, Levitt, and Stelzner 86] Describes the use of multiple worlds in the analysis of contingencies in project plans.

[Lalo 88] Presents TIBRE an expert system for maintaining the consistency in a rule base.

[Latombe 79] Discusses a JTMS-like system that learns from its failures.

[Mason, Searfus, and Lager 88a], [Mason, Johnson, Searfus, Lager, and Canales 88b] Describes a system that interprets seismic data and uses a TMS.

[Mavrovouniotis and Stephanopoulos 87] Discusses the use of a TMS on a system that reasons about orders of magnitude.

[Provan 88e] Describes a system that uses TMS to model-based vision.

[Nado and Fikes 87] Discusses the role of a TMS in a frame language with inheritance.

[O'Rorke 83] Describes how non-monotonic dependencies of a JTMS can be used for story processing.

[Park 87] Discusses a belief theory which is closely related to an acting model.

[Provan 87] Discusses the efficiency of the ATMS in solving scene representation problems.

[Provan 88e] Describes a system that uses TMS to model-based vision.

[Rose 88a, 88b, 90] Discusses the role of a TMS in the process of discovery.

[Rose and Langley 86a] Describes STARf a system for inferring components of chemical substances based on ATMS.

[Schor 86] Describes the use of a TMS in an expert system implemented in OPS5.

[Shimazu and Takashima 89] Describes the use of a TMS in the task of memory retrieval.

[Shrobe 79] Discusses how REASON, a system that uses JTMS, reasons about side-effects in programming.

[Steel 87] Discusses dependency-directed backtracking and relates it with the JTMS.

[Steele, Richardson, and Winchell 89] Describes a system for the design of circuits that uses ATMS.

[Vilain 82] Describes a system for reasoning about time that embodies a mechanism similar to a TMS.

[Wellman and Simmons 88] Describes SERF, the set reasoning facility, based on LTMS.

[Worden, Foote, Knight, and Andersen 86] Describes an architecture of co-operative expert systems and points out the role that a TMS may play in it.

**Acknowledgements**

I would like to thank Maria R. Cravo for her help and fruitful discussions during compilation of this bibliography. This work was partially supported by Junta Nacional de Investigacao Cientifica e Tecnologica (JNICT), under Grant 87-107.

**References**

Items marked with * have not been inserted in the body of the text. These are references which have been made to systems that use TMS techniques but I haven’t checked the paper (because I don’t have them).


de Kleer J., “Problem-Solving with the ATMS”, Artificial Intelligence 28, No.2, pp.197-204, 1986c.


Doyle J., “A Truth Maintenance System”, Artificial Intelligence 12, No.3, pp.231-272, 1979. Also in Readings in Artificial Intelligence, Webber and Nils-


Koff C.N., “A Specialized ATMS for Equivalence Relations”, Proc. Sev-


Perlis D., “A Bibliography of Literature on Non-Monotonic Reasoning”, in Readings in Nonmonotonic Reasoning, Ginsberg (ed.), pp.466-477, Los Altos,


Reinfrank M., “Reason Maintenance Systems"
Articles


About the Author
Joao P. Martins is Associate Professor of Computer Science at the Technical University of Lisbon. He received his Ph.D. in Artificial Intelligence from the State University of New York at Buffalo. His interests include belief revision systems, knowledge representation, and reasoning. He is the author of Introduction to Computer Science published by Wadsworth in 1989.