X-tra as a toolbox for truth maintenance

François Charpillet, Pierre Marquis, Jean-Paul Haton

UNITÉ DE RECHERCHE
INRIA-LORRAINE

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P. 105
78153 Le Chesnay Cedex
France
Tél.: (1) 39 63 55 11

Rapports de Recherche

N° 1456

Programme 3
Intelligence artificielle, Systèmes cognitifs et Interaction homme-machine

X-TRA AS A TOOLBOX FOR TRUTH MAINTENANCE

François CHARPILLET
Pierre MARQUIS
Jean-Paul HATON

Juin 1991

X-TRA as a Toolbox for Truth Maintenance

François Charpillet, Pierre Marquis et Jean-Pierre Haton
CRIN/INRIA-Nancy, BP 239
54506 Vandœuvre-lès-Nancy Cedex, France

abstract :

X-TRA is a development tool for knowledge-based systems. It integrates a toolbox for truth maintenance based on both TMS and ATMS techniques. In this paper we present facilities provided in X-TRA for the truth maintenance task.

X-TRA, une boîte à outils pour le maintien de cohérence dans les bases de connaissances

François Charpillet, Pierre Marquis et Jean-Paul Haton
CRIN/INRIA-Nancy, BP 239
54506 Vandœuvre-lès-Nancy Cedex, France

résumé :

X-TRA est un outil de développement de systèmes à bases de connaissances. Il comporte une "boîte à outils" pour le maintien de cohérence fondé à la fois sur les techniques de TMS et d'ATMS. Dans ce papier, nous présentons les principales fonctionnalités proposées par X-TRA pour cette tâche de maintien de cohérence.

N° de programme INRIA : 3 (Intelligence artificielle, sciences cognitives et interaction homme-machine).

X-TRA as a Toolbox for Truth Maintenance

F. Charpillet, P. Marquis et J. P. Haton
CRIN / INRIA-Nancy
Campus Scientifique - BP 239
54506 Vandoeuvre lès Nancy Cedex
FRANCE

1 INTRODUCTION

X-TRA[1] is a development tool for knowledge-based systems. It integrates a toolbox for truth maintenance based on both TMS and ATMS techniques. In this paper we present some facilities provided in X-TRA for the truth maintenance task.

2 OVERVIEW OF X-TRA

2.1 Basics of Truth Maintenance

Most knowledge-based systems which have been developped so far operate in a closed world with restricted communication with their environment. However, the use of these systems in real industrial environments makes it necessary to consider more sophisticated interactions with the external world. This implies that the inference mechanisms must be able to cope with complex requirements such as incompleteness of data and temporal evolution. The processing of incomplete information (temporal or not) imposes the evaluation of different assumptions (eventually competitive). The temporal evolution can further invalidate some assumptions and thus questions part of the deductions. Therefore we need recording data dependencies in order to provide an automatic maintenance of the knowledge base when evolutive information have to be handled.

A solution consists in designing a truth maintenance system (TMS). The aim of a TMS is to record each inferential step processed by the reasoning

[1] The development of this projet was partly supported by the french programming naval center (CPM) and has also involved Cognitech a French hightech company specialized in A.I. This project was one of the starting point of the AITRAS Esprit Project.

module (RM) and the hypotheses drawn to produce them. The inferential steps are transmitted to the TMS in an uniform form, i.e. a justification. Each time the RM produces a new justification, the TMS provides the set of deductions (context) that RM can further use for new inferences. Computing a context consists in determining a consistent subset of the deductions depending on a consistent set of assumptions.

Different kinds of truth maintenance systems have been developped depending on:

- the kind of justifications (monotonic or non monotonic),
- the ability of taking into account one or several consistent subsets of all the data known by the system,
- the use of explicit or implicit assumptions,
- the logical status of an information : (true, false) or (true, unknown)

Two main approaches can be considered, the first one with backtracking mechanisms (JTMS or NMJTMS) (Doyle, 1979), the other one with parallel processing of hypotheses (ATMS) (De Kleer, 1988).

In Doyle's approach the various alternatives associated with hypothetical data are considered and tested sequentially. The justifications can be non monotonic (in this case the truth maintenance system is an NMJTMS), and the TMS maintains *only* one consistent subset of the database (one context). The module of truth-maintenance is called upon each time a contradiction is detected and when a new justification is provided by the inference engine. Justifications record dependencies between data. They have to be taken into account by the TMS for associating a state IN or OUT with each fact. TMS also determines the minimal subset of hypotheses to be cancelled when it is necessary to restore consistency, usually by using a dependency-directed backtracking technique. The main drawback of this approach lies in its inhability in evaluating different competing solutions.

From a chronological point of view, ATMS-like systems were introduced much later. The need for ATMS appeared after extensive experiences with problem solvers based on JTMS. JTMS has the fundamental limitation that it constraints the problem solver to reason in only one context at a time. In an ATMS-based problem solver, all the possible consistent sets of data can be considered at any time. Justifications are monotonic, but many efforts aim at extending their capabilities to include non-monotonicity (Dressler, 1988) such as the one we are presenting in this paper. Several systems based on ATMS technique have been developped. For instance, the industrial systems ART ™ (Clayton, 1985) and KEE ™ (Morris & Nado, 1986) integrate mechanisms closed to De Kleer's ATMS.

2.2 Basics of X-TRA

A justification j as introduced in section 2.1 can be non-monotonic :

$$j : a_1, ..., a_n \ OUT(b_1, ..., b_m) \rightarrow c$$

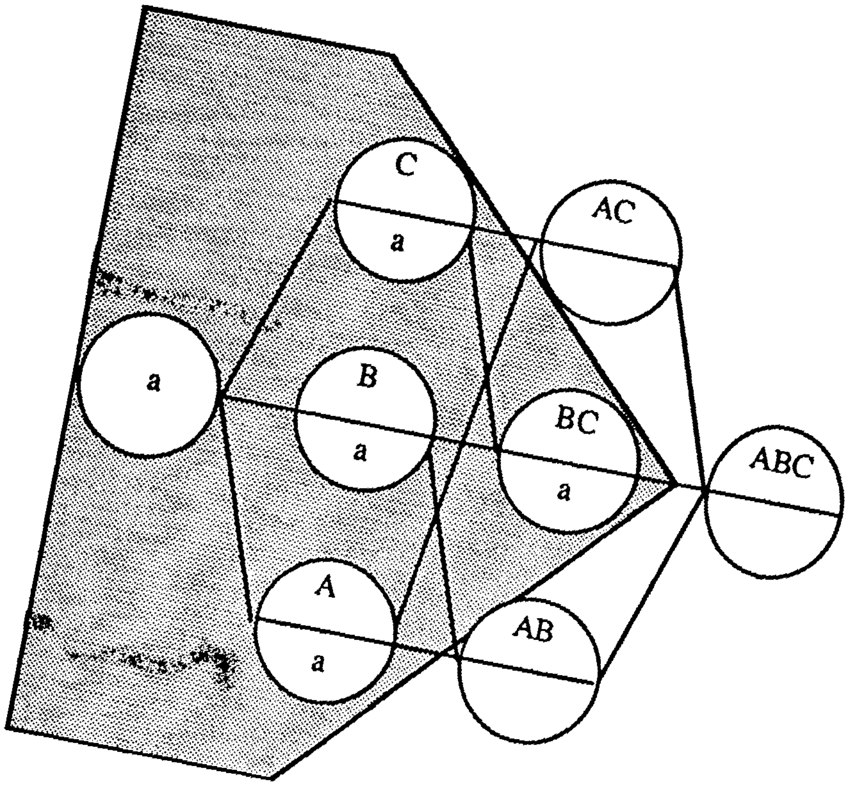
where $a_1, ..., a_n, b_1, ..., b_m$, c are nodes. We define antecedent_IN(j) as $\{a_1, ..., a_n\}$, antecedent_OUT(j) as $\{b_1, ..., b_m\}$and consequent(j) as c. Nodes $a_1, ..., a_n, b_1, ..., b_m$ are associated with facts (from the set F) or hypotheses (from the set H) and c is always associated with a fact or the symbol $\perp$.

Such a justification j can be interpreted as "c holds when $a_1, \ldots, a_n$ hold and no b_i among $b_1, \ldots, b_m$ holds".

Justifications have to be considered by the truth maintenance module of X-TRA in order to compute the domain of validity of each node n, i.e. the set of all environments $\{E_i(n)\}$ in which n holds. $E_i(n)$ is a set of hypotheses (a subset of the power set $\mathcal{P}(H)$) from which n can be derived using the set J of justifications.

The basic structure in X-TRA is called *interval*. An interval I is a couple $[E_k \mid E_{k1}, \ldots, E_{kp_k}[$ where E_k is its inf-part and $E_{k1}, \ldots, E_{kp_k}$ is its sup-part. We represent the set $\{E_i(n)\}$, by a set of intervals (or label) LABEL(n) = $\{[E_k(n) \mid E_{k1}, \ldots, E_{kp_k}[\}$ such that, for each i, there exists k for which $E_i(n) \supseteq E_k(n)$, $E_i(n) \not\supseteq E_{kj}$ and n does not hold in E_{kj} for each $j \in [1, \ldots, p_k]$. The membership of a fact n to an interval $\{[E_k(n) \mid E_{k1}, \ldots, E_{kp_k}[\}$ can be interpreted as "n holds from $E_k(n)$ to $E_{k1}, \ldots, E_{kp_k}$". An interval $[E_k \mid E_{k1}, \ldots, E_{kp_k}[$ represents every environment E_i such that $E_i \supseteq E_k$ and $E_i \not\supseteq E_{kj}$ for each $j \in [1, \ldots, p_k]$.

Example The membership of node a to the interval $[\varnothing \mid \{A, B\}, \{A, C\}[$ is illustrated on figure 1. The environments associated with this interval are hatched.



The set of environments in which a holds is defined by an interval.

Figure 1. Membership to an interval.

When two intervals I_1 and I_2 define the same set of environments, they are considered as equivalent : $I_1 \approx_{int} I_2$. If the set of environments represented by I_1 is included in the set of environments represented by I_2, we note $I_2 \supseteq_{int} I_1$. For instance, in the previous example, intervals $[\ \varnothing \mid \{A, B\}, \{A, C\}[$ and $[\ \varnothing \mid \{A, B\}, \{A, C\}, \{A, B, C\}[$ are equivalent and interval $[\ \varnothing \mid \{A, B\}, \{A, C\}[$ is included in interval $[\ \varnothing \mid \{A, B\}[$. In the equivalence class of interval I, SIMP(I) denotes the simplest interval, i.e. the one for which the sup-part is minimal for the set inclusion. By extension, we also note $L_1 \approx_{lab} L_2$ when labels L_1 and L_2 represent the same set of environments and $L_2 \supseteq_{lab} L_1$ when the set of environments represented by L_1 is included in the set of environments represented by L_2.

Before describing the truth maintenance functionalities offered by X-TRA, we have to define three operators on the set of all labels on H, $LABELS_H$, i.e. union, intersection and complementation. These operators allow to compute respectively the union, intersection and complementation of sets of environments from their representations based on sets of intervals. It is necessary to define first the corresponding operators from intervals to labels.

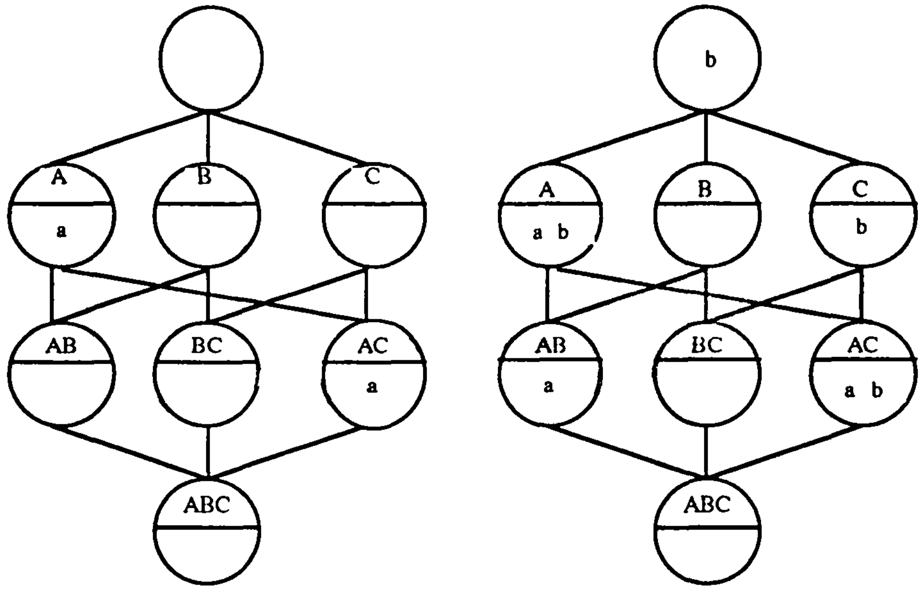
Let $[E_{k1}, ..., E_{kp_k}[$ and $[E_l \mid E_{l1}, ..., E_{lp_l}[$ be two intervals :

• their union, noted $[E_k \mid E_{k1}, ..., E_{kp_k}[\cup_{int} [E_l \mid E_{l1}, ..., E_{lp_l}[$ is the set $\{[E_k \mid E_{k1}, ..., E_{kp_k}[, [E_l \mid E_{l1}, ..., E_{lp_l}[\}$;

• their intersection, noted $[E_k \mid E_{k1}, ..., E_{kp_k}[\cap_{int} [E_l \mid E_{l1}, ..., E_{lp_l}[$ is the set $\{SIMP([(E_k \cup E_l) \mid E_{k1}, ..., E_{kp_k}, E_{l1}, ..., E_{lp_l}[)\}$;

• the complement of the interval $[E_k \mid E_{k1}, ..., E_{kp_k}[$ in $LABELS_H$, noted $\mathcal{C}_{int}([E_k \mid E_{k1}, ..., E_{kp_k}[)$, is the set $\{[\varnothing \mid E_k[, [E_{k1} \mid [, ..., [E_{kp_k} \mid [\}$.

The interval $[E \mid [$ is such that inf-part$([E \mid [) = \{E\}$ and sup part$([E \mid [) = \varnothing$. Union, intersection and complementation of intervals are illustrated on figure 2.



Interval in which a holds = $[\{A\} \mid \{A,B\}, \{B,C\}[$

$\mathcal{C}_{INT}([\{A\} \mid \{A,B\}, \{B,C\}[)=$
$\{[\{A,B\} \mid [, [\{B,C\} \mid [, [\varnothing \mid \{A\} \mid [\}$

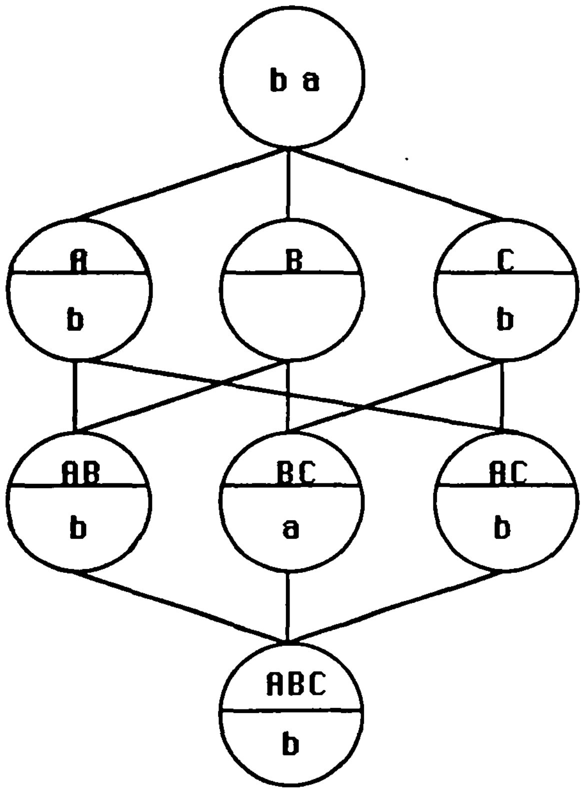
Interval in which a holds : $I1=[\{A\} \mid \{B,C\}[$
Interval in which b holds : $I2=[\varnothing \mid \{B\}[$
Interval in which a and b holds :
$I1 \cap_{int} I2= \{[\{A\} \mid \{B\}[\}$
Intervals in which a or b hold :
$I1 \cup_{int} I2 = \{[\varnothing \mid \{B\}[, [\{A\} \mid \{B,C\}[\}$

Figure 2. Operators on intervals.

Let $\{[E_k \mid E_{k1}, \ldots, E_{kp_k}[\}$ and $\{[E_l \mid E_{l1}, \ldots, E_{lp_l}[\}$ be two labels of LABELS_H:

- their union, nöted $\{[E_k \mid E_{k1}, \ldots, E_{kp_k}[\} \cup_{lab} \{[E_l \mid E_{l1}, \ldots, E_{lp_l}[\}$ is the set $\{[E_k \mid E_{k1}, \ldots, E_{kp_k}[\} \cup \{[E_l \mid E_{l1}, \ldots, E_{lp_l}[\}$;
- their intersection, noted $\{[E_k \mid E_{k1}, \ldots, E_{kp_k}[\} \cap_{lab} \{[E_l \mid E_{l1}, \ldots, E_{lp_l}[\}$ is the set $\bigcup_{lab\ (\text{for all } k,\ l)} ([E_k \mid E_{k1}, \ldots, E_{kp_k}[\cap_{int} ([E_l \mid E_{l1}, \ldots, E_{lp_l}[)$;
- the complement of the label $\{[E_k \mid E_{k1}, \ldots, E_{kp_k}[\}$, noted $\mathcal{C}_{lab}(\{[E_k \mid E_{k1}, \ldots, E_{kp_k}[\})$, is the set $\bigcap_{lab\ (I \in \{[E_k \mid E_{k1}, \ldots, E_{kp_k}[\})} (\mathcal{C}_{int}(I))$.

Union, intersection and complementation of labels are illustrated on figure 3.



$\mathrm{Label}(b)=\{[\varnothing \mid \{B\}[, [\{A, B\} \mid [\}$
$\mathrm{Label}(a)=\{[\varnothing \mid \{A\}, \{B\}, \{C\}[, [\{B, C\} \mid \{A, B, C\}[\}$
$\mathrm{Label}(a) \cup_{lab} \mathrm{Label}(b) = \{[\varnothing \mid \{B\}[, [\{A, B\} \mid [, [\{B,C\} \mid [\}$
$\mathrm{Label}(a) \cap_{lab} \mathrm{Label}(b) = \{[\varnothing \mid \{A\}, \{B\}, \{C\}[\}$
$\mathcal{C}_{lab} (\{[\varnothing \mid \{B\}[, [\{A, B\} \mid [\})= \{[\{B\} \mid \{A, B\}[\}$

Figure 3. Operators on the set of all labels.

We will now consider three kinds of justifications and briefly present some properties of the derivability relation based on them.

We first consider how the truth maintenance module works when the set of justifications J contains only monotonic justifications $j : a_1, ..., a_n \rightarrow c$. In this case, the node n holds in an environment E, noted $E, J \vdash n$, if there exists a sequence $(S_i) = S_0, ..., S_m$ for which :

- $S_0 = E$;
- $S_{i+1} = S_i \cup \{c_l \mid \text{there exists } j_l : a_{l1}, ..., a_{ln_l} \rightarrow c_l \text{ in } J \text{ and } S_i \supseteq \text{antecedent\_IN}(j_l)\}$;
- $\perp \notin S_i$ for each $i \in [1, ..., m]$;
- $n \in S_m$.

Since $\vdash$ is a monotonic relation, if $E, J \vdash n$ then $E_i, J \vdash n$ for each $E_i \supseteq E$. So LABEL(n) is uniquely determined and can be represented by the union of intervals $\{[E_{min_i} \mid [\}$ where E_{min_i} is a minimal element of $\mathcal{P}(H)$ in which n holds. Hence, when J consists of monotonic justifications, the truth maintenance module of X-TRA works like a standard ATMS.

Let us now consider how this module works when the set of justifications J contains also non-monotonic justifications $j : a_1, ..., a_n, \text{OUT}(b_1, ..., b_m) \rightarrow c$. In this case, as in a TMS, more than one labelling of nodes is possible in general. The resulting labelling $L = \{\text{LABEL}(n) \mid n \in F\}$ depends on the ordering representing how the justifications are provided by the inference engine.

The node n holds in an environment E for a labelling L, noted $E, J \vdash_L n$, if E is an element of the set of environments described by LABEL(n) in L. Clearly, $\vdash_L$ is a non-monotonic relation. In order to test whether $E, J \vdash_L n$, it is necessary to compute the following sequence $(S_i) = S_0, ..., S_m$:

- $S_0 = E$;
- $S_{i+1} = S_i \cup \{c_l \mid \text{there exists } j_l : a_{l1}, ..., a_{ln_l}, \text{OUT}(b_{l1}, ..., b_{lm_l}) \rightarrow c_l$ in J and $S_m \supseteq \text{antecedent\_IN}(j_l)$ and for each b_{lk} in antecedent_OUT(j_l), $b_{lk} \notin S_m\}$;
- $\perp \notin S_i$ for each $i \in [1, ..., m]$;
- for each node p in S_m, E(n) is an element of the set of environments described by LABEL(p) in L ;
- $n \in S_m$.

This computation does not always terminate[1] as it will be illustrated in section 3.

However, a particular subset of non-monotonic justifications is interesting : when each j_l in J is such that $H \supseteq \text{antecedent\_OUT}(j_l)$. In this case, the computation of (S_i) always terminate and the labelling L is unique : the truth maintenance module operates like O. Dressler's extended ATMS (Dressler, 1989).

[1] The sequence (S_i) is computed by application of an operator O such that $S_{i+1} = O(S_i)$. This sequence is finite when the set $S_i = S_m$ under consideration is a fixed point for this operator. Note that O does not always admit a fixed point. In such a case, the computation of (S_i) does not terminate.

3 USING X-TRA

In this section, we refer to the classical examples of even and odd loop problems in order to illustrate how X-TRA can be used to perform truth maintenance tasks. In a first time, we present how X–TRA can be used as a JTMS. In a second time, we present how X–TRA can be used as an extended ATMS.

3.1 Using X-TRA as a JTMS

The Even Loop Problem

Let us consider the justifications J = {OUT(a) → b; OUT(b) → a} where F ⊇ {a, b}. There exists two possible labellings depending on the ordering representing how the justifications are taken into account by the truth maintenance module. Only one of them is constructed. For instance, if the first justification OUT(a) → b is considered before the second one, then L_1 = {LABEL(b) = {[∅ | [ ]} else L_2 = {LABEL(a) = {[∅ | [ ]}.

The Odd Loop Problem

Let us consider the justifications J = {OUT(a) → b; b → a} where F ⊇ {a, b}. The computation of the labelling does not terminate.

3.2 Using X-TRA as an extended ATMS

The Even Loop Problem

Let us consider the justifications J = {OUT(A) → b; OUT(B) → a; OUT(A), a → ⊥; OUT(B), b → ⊥} where F ⊇ {a, b} and H ⊇ {A, B}. There exists one labelling corresponding to the two labellings provided in 3.1 for the same problem. This labelling L = {LABEL(a) = {[∅ | B[ ]}, LABEL(b) = {[∅ | A[ ]}, given by the truth maintenance module, appears at figure 4.

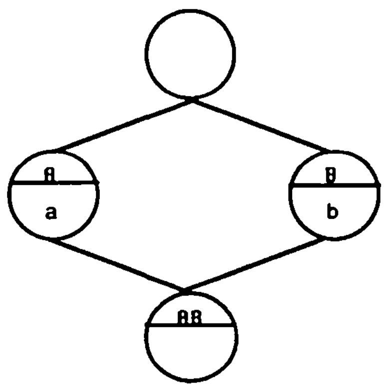


Figure 4. Even Loop.

The Odd Loop Problem

Let us consider the justifications $J = \{OUT(A) \rightarrow b; b \rightarrow a; OUT(A), a \rightarrow \perp\}$ where $F \supseteq \{a, b\}$ and $H \supseteq \{A\}$. The computed labelling L is empty.

4 CONCLUSION

We have presented in this paper the truth maintenance functionalities of X-TRA, a tool for the development of knowledge-based systems with hypothetical reasoning. X-TRA can be viewed as a toolbox for truth maintenance based on TMS and ATMS techniques. The truth maintenance module of X-TRA is closely integrated with an inference engine using RETE (Forgy, 1982) and TREAT (Miranker, 1987) compilation algorithms. It has also been implemented in ATOME a blackboard tool developed by our group (Laasri *et al.*, 1988). Moreover, an ART ™ compatible mode is also available. X-TRA offers the possibility to encode J. Doyle's TMS, J. De Kleer's ATMS, several extended ATMS as J. De Kleer's NATMS (De Kleer, 1988) and O. Dressler's model.

X-TRA is operational on Explorer Lisp machine and SUN workstation. It is used in various applications including the ESPRIT project AITRAS for the real time interpretation of industrial signals.

5 BIBLIOGRAPHY

B. D. Clayton, [1985] "A First Look at Viewpoints", Inference Corporation.

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

O. Dressler, [1988] "Extending the Basic ATMS", Proceedings of ECAI, 535-541, Munich.

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

J. de Kleer, [1988] "A General Labeling Algorithm for Assumption-Based Truth Maintenance", Proceedings of AAAI, 188-192.

L. Forgy, [1982] "RETE : A Fast Algorithm for the Many Pattern/Many Object Pattern Matching Problem", Artificial Intelligence 19 : 17-37.

H. Laasri, B. Maitre, T. Mondot, F. Charpillet, J.P. Haton, [1988] "ATOME : a Blackboard Architecture with Temporal and Hypothetical Reasoning", Proceedings of ECAI, 1-5, Munich.

P. Miranker, [1988] "TREAT : A New and Efficient Match Algorithm for AI Production Systems", Ph.D. Thesis, Computer Science Dept., Columbia University.

P.H. Morris, R.A. Nado, [1986] "Representing Actions with an Assumption-Based Truth Maintenance System", Proceedings of AAAI, 13-17.

ISSN 0928-6888