

A Modal Temporal Logic and its Models Underlying Variants of Planning Algorithms

Klaus P. Jantke

Fachbereich IMN
HTWK Leipzig
04277 Leipzig, Germany

Oksana Arnold

Inst. f. Wirtschaftsinformatik
Universität Leipzig
04109 Leipzig, Germany

Abstract

The focus of this paper is on the introduction of some modal temporal logic and its application to knowledge processing under the particularly difficult circumstances of disturbed complex dynamic processes. The development of this logic has been driven by the overall approach to knowledge-based process supervision and control. Within this comprehensive approach, therapy control is playing a crucial role. The therapy control concept comprises the generation of therapy or repair plans, plan execution and constraint monitoring and includes plan revision. The incompleteness and vagueness of available information about an accidentally disturbed process requires a particularly tailored logical approach. The basic concepts are introduced and exploited for developing a collection of planning algorithms properly exceeding previous approaches.

1 Introduction

The present paper focusses on a modal temporal logic and its use for planning in complex dynamic environments. Although there are implemented two planning systems under SUN/OS on Sparc Stations (in Allegro Common Lisp, cf. [2] and [3], resp. Quintus Prolog, cf. [22], [21] and [18]) which nicely demonstrate the feasibility of the concepts developed, this paper is more theoretical in spirit. It is intended to discuss certain novel theoretical issues. As an immediate outcome of our approach, there will be a collection of novel variants of planning algorithms.

The paper aims at explaining and illustrating the modal temporal logic and its use for therapy plan generation. Constraints are verified (more precisely: refuted) and monitored with respect to some dynamically changing knowledge base. Particular emphasis is put on the problem that the available information about underlying dynamic processes may change during knowledge processing and, thus, may fool the automated reasoning procedures. The framework allows to extract essentially different versions of plan generation to cope with the process dynamics.

In the present paper, technicalities are suppressed as much as possible. Some emphasis is put on a careful motivation particularly stressing the peculiarities of our characteristic applications and on the essential problems and ideas of consistent planning.

[16] and [17] are two recent papers of us closely related to the present one. In contrast to the present one, [16] does not emphasize logic at all. It is clearly focussed on the problem of communication between the therapy control module (planning, in particular) and the simulation module. Although [17] is also putting some emphasis on logic, it deals with very specific phenomena like monotonicity, for instance.

1.1 Motivation

The application domain is therapy or repair plan generation for complex dynamic systems. Therapy planning is invoked, when a target process is seriously disturbed such that conventional process control fails to drive the process back into an acceptable working regime. Knowledge-based process supervision and control necessarily depends on representing and processing time information. As therapy plans are designed for execution in the future, this is particularly important and difficult. Before going into details, one should recall that the following aspects have an essential impact on the present approach to plan generation:

- I Complex dynamic systems like chemical processes, for instance, have a score or more peculiarities which have to be taken into account for developing, implementing and applying appropriate planning concepts.
- II Planning is integrated into a comprehensive approach to knowledge-based process supervision and control. Thus, planning has to respect the surrounding knowledge processing conditions, and it has to interact smoothly with the other knowledge processing components.

Within the present paper, we provide only a necessary minimum of preliminaries to reserve sufficient space for the logic and its applications. The peculiarities of the class of our target processes has been intensively discussed in [7], e.g. The concept of knowledge-based process supervision and control has been advocated in several of our publications. [23] and [10] are the seminal papers of our group within the WISCON project on *Knowledge-Based Process Supervision and Control*.

Under the assumption of some appropriately chosen modal temporal logic, i.e. some formal language, some corresponding model concepts for defining validity, and some operational formalization of entailment, the paper focusses on questions like the following:

- Q1 How to describe appropriately the difficulty of process knowledge which changes dynamically during the reasoning process?
- Q2 How to describe the resulting problems sufficiently clear, to allow for appropriate investigations and algorithmic solutions?
- Q3 What are the key approaches to cope with dynamically changing knowledge underlying therapy plan generation?

Chapter 4 is intended to answer these questions by demonstrating the use of our underlying logic. Some variants of plan generation within the same framework, but from a different perspective, are investigated in [14] and [18].

The remaining part of chapter 1 is aimed at setting the stage for the target investigations.

1.2 Peculiarities of Disturbed Complex Dynamic Systems

From the huge amount of peculiarities of complex dynamic systems (cf. [7] for a comprehensive discussion which meets our needs quite well), the following are particularly relevant to understand the difficulties of generating therapy plans in a timely response to disturbances and diagnosed primary faults.

- Values of process parameters may be set only by indirect influence, usually. Consequently, the effects of actions are not always definite, i.e. formal postconditions may not exist.
- Sometimes, parameters which are necessary to describe the complete state of some device are not accessible (measuring may be very time consuming, e.g.).
- The success of actions depends, in general, on the availability of technological resources (preconditions), like energy, conveyance pipes and storage capacity.
- There may be some need for simultaneous actions resulting from the structure of the equipment and the running processes.

These are characteristics of the class of target processes. Furthermore, there are other important properties which are less peculiar, but which interfere with properties like above. Such an interference may cause additional difficulties. Here, we do not go into further detail (cf. [7]).

As a consequence, STRIPS-like planning approaches (cf. [11]) may turn out to be inappropriate. Thus, in particular, a planning approach for real-time generation of therapy plans needs to be based on certain innovative concepts. Moreover, the above peculiarities

will have a serious impact on the overall knowledge processing approach. Among others, they will be reflected by the knowledge representation and by basic methods of knowledge management. This, again, is fundamental for embedding therapy plan generation appropriately into the overall approach.

1.3 Knowledge-Based Process Supervision and Control

Essentially, knowledge-based process supervision and control (cf. [23] and [10]) is understood as an extension of a conventional process supervision and control system. It is, so to call, an intelligent add-on. Its ultimate goal is to invoke knowledge-based methods exactly if conventional methods of process supervision fail.

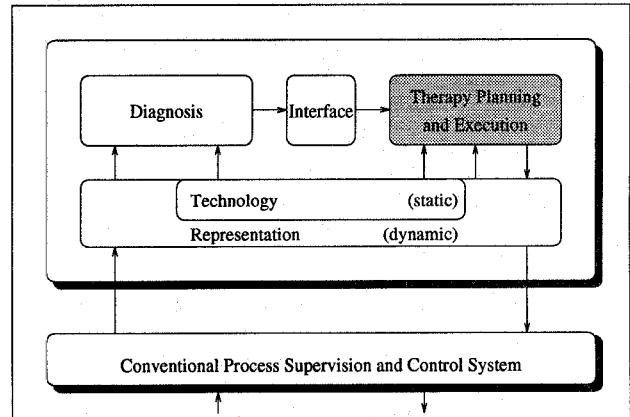


Figure 1: Simplified Overall System Architecture

The figure is showing a quite simplified architecture. It contains only the minimal components relevant for the problem of therapy plan generation. More advanced approaches (cf. [16], e.g.) may require further knowledge processing components, like a simulation module, for example. The so-called technology representation is the system's knowledge base which basically contains

- **static process knowledge** like classification hierarchies, aggregation hierarchies, and topological structure (We do not go into detail, cf. [5].),
- **static inference knowledge** which may essentially depend on the peculiarities of the inference mechanisms in use (For example, some module for diagnosis may be based on a hierarchy of process models, and it may build its diagnoses via construction of causal nets from a reservoir of causal nodes.),
- **dynamic process information** about the past, the present and, perhaps, the future of the target process which – in all cases when the system is invoked – is suffering from some disturbance.

The logic developed in the sequel describes a formal approach for dealing with the dynamic process information.

1.4 The Basic Planning Approach

The underlying plan generation approach has been published in [6], [7], [14], and [13], for instance. Essentially, plans are finite, directed and acyclic graphs. Nodes are describing actions which may be either elementary or compound. Elementary actions have a well-defined operational semantics. In our implementations (cf. [2], [3], [22], [21], and [18]), these semantical concepts are formalized by rule bases. This allows reactivity through condition monitoring at execution time. For compound nodes, there are alternative graphs listed which may be used for substitution. Thus, plan generation is mainly node expansion. Roughly speaking, we have an underlying graph grammar for therapy plan generation. The graph-theoretic approach is described in [8], [12], and [9]. There is a detailed mechanism of guarded graph rewriting. The resulting formal language of graphs is known to have highly interesting properties (cf. [19] and [20]). Here, we are not going into details.

For the present paper, the following perspective is essential: For node replacement, one has to check certain constraints attached to the graphs possibly substituted. Constraint checking depends on time information and, naturally, on data about the process dynamics. This requires

- a formal language for representing constraints,
- a model concept to determine validity, and
- an implementation of the entailment operator.

This logic briefly described in chapter 2 is partially inspired by [1] as well as by [24]. Semantical concepts are somehow similar to those in [25], although the treatment there is not very formal. However, our approach exceeds the three mentioned considerably.

1.5 Peculiarities of Therapy Plan Generation for Disturbed Dynamic Processes

We conclude our introduction with pointing to two essentials of our planning approach:

- Plans are programs to be executed on the underlying installation. Therefore, plan generation is program synthesis. More precisely, due to the unavoidable incompleteness of information, therapy plan generation for complex dynamic processes is inductive program synthesis (cf. [6] and [15], in particular).
- Due to the process dynamics and to the incompleteness of information, executability of plans resp. programs is undecidable, in advance. Thus, one needs another concept to distinguish "good" plans. This is the concept of consistency to be introduced below.

Now we are going to introduce some logic which has been published so far in German, only (cf. [4]).

2 A Modal Temporal Logic

We introduce a logical approach to representing and processing incomplete knowledge about dynamically changing processes. There are the following fundamental properties of our logic (cf. [4]):

- The crucial logical reasoning process within the developed planning approach is *constraint refutation*.
- A *more comprehensive* knowledge base allows to refute *more* constraints.
- The *complexity of constraint refutation* is growing with the number of known episodes, i.e. with the completeness of histories.
- There is an *equivalence concept* for histories which allows to simplify knowledge representation.
- The *complexity of graph rewriting* is growing with the number of refutable constraints.

As a consequence, there is a tradeoff between efficiency and accuracy of logical reasoning.

2.1 Syntax

There are 5 basic sorts, 2 non-temporal and 3 temporal sorts. First, there are the sorts of numeric resp. symbolic process parameters. These sorts have a large collection of application-specific constants. Furthermore, there are the usual operators of arithmetic and elementary set theory suppressed here. The three temporal sorts are representing *time points*, *time durations*, and *time intervals*. There are several operators relating these sorts. The sort of time points and the sort of time durations have constants, whereas the sort of time intervals has not. There is a particular operator *exp* for generating time intervals by two time points. Thus, a ground term of the sort of time intervals looks like *exp*(\$95.12.01.09-52-05,\$95.12.01.09-53-40). Another version is *exp*(\$95.12.01.09-52-05,\$95.12.01.09-52-05+\$95s), where the second argument term is built from two constants of the sorts of time points and time durations connected by addition.

Upon the non-temporal part of this signature, there is built a standard first order predicate language. The resulting formulae of classical predicate logic are called *constraints*.

If φ is any constraint and i is any term of the sort of time intervals, then (i, φ) is called a *formula*. All formulae must be of this structure. There is no way to assign truth values to constraints. Only formulae may be either valid or not in dependence on some underlying knowledge.

For a large application area (cf. [26] for an acquisition, formalization and representation of therapy knowledge in a complex chemical installation, for instance), a subclass of formulae is sufficient where no iteration of quantifiers occurs. We call these formulae *technological constraints*.

2.2 Models and Validity

For the temporal sorts, there is assumed some standard semantics: Time points are interpreted as natural numbers and time durations are interpreted as integers. As known from most Lisp systems, the natural number 0 means the real time point of 0:00 on January 1, 1900. The corresponding constant is written \$00.00.00-00-00.

The basic semantical concepts are called *episodes* and *histories* describing process parameters. The approach in [25] is adopted and properly refined. An episode is a rectangle spanned by a time interval and a value interval. Histories are finite collections of episodes which do not overlap in time. There is an arithmetic of histories. The operations reasonably propagate the uncertainty of information. (Figure 3 contains some histories, for illustration.)

Models are finite collections of histories containing for every non-temporal constant its uniquely defined history. Motivated by our underlying knowledge-based system approach, (the dynamic parts of) technology representations are taken as models and denoted by TR , for short.

The modal character of our logic is determined by its validity concept. For every predicate symbol, there is an appropriate interpretation taking the uncertainty of information into account. For readability, we prefer an illustration to a complete formalization: Assume any two ground terms x and y of the same non-temporal sort and any ground term $i = \exp(t_1, t_2)$ denoting any time interval. Furthermore, assume any technology representation with histories h_x and h_y interpreting the constants (meaning process parameters) x and y , respectively. It holds, for example,

$$TR \models (\exp(t_1, t_2), x < y)$$

exactly if the episodes (i.e. rectangles) in h_x and h_y resulting from a restriction to the interval $[t_1, t_2]$ of natural numbers are not disjoint. Here, each t_n is the natural number interpreting the corresponding ground term tn . In this case, at every moment in time between t_1 and t_2 , there is an opportunity to satisfy the given technological constraint. Thus, the corresponding formula is *true* in the model.

2.3 Application

This paragraph is intended to sketch the use of deduction in therapy plan generation.

Therapy knowledge is represented as a collection of action scripts. In formal terms, action scripts are graphs. To compound nodes of so-called compound action scripts, there are assigned certain other action scripts for possible substitution. Node expansion during planning is guarded by technological constraints attached to the action scripts considered for potential substitution.

The basic notations are as follows: G_m is some graph describing an incompletely developed plan and c is one of its compound nodes. For the node c , there are some alternative graphs for substitution. Every such graph G_n is accompanied by some technological

constraint φ_n . (In fact, there are start constraints and interval constraints which are treated differently during plan execution. But we do not go into these details, here.) The result of replacing c by G_n is denoted $G_m[c \leftarrow G_n]$.

In the algorithmic approach developed, node expansion is only permitted in case all preceding nodes are already elementary. As a result, for every ongoing node expansion there exists some time interval estimating the possible start time of the corresponding compound action. If i is any term describing this time interval under consideration, the planning step from G_m to $G_m[c \leftarrow G_n]$ is admissible exactly if

$$TR \not\models (i, \varphi_n)$$

holds with respect to the underlying model TR . Thus, *constraint refutation* is the crucial deductive task of our therapy plan generation approach.

Refutability of technological constraints over finite models is decidable.

3 Variants of Consistent Algorithms for Therapy Plan Generation

Besides motivating and introducing some modal temporal logic and using this particular logic for constraint refutation within plan generation, the focus of the present paper is on exploitation of this logical approach towards the development of a novel collection of planning algorithms. In [14] and [18], there have been introduced a few variants of planning algorithms. These variants were driven by investigating special heuristics. In contrast, the present investigation is clearly driven by logical phenomena.

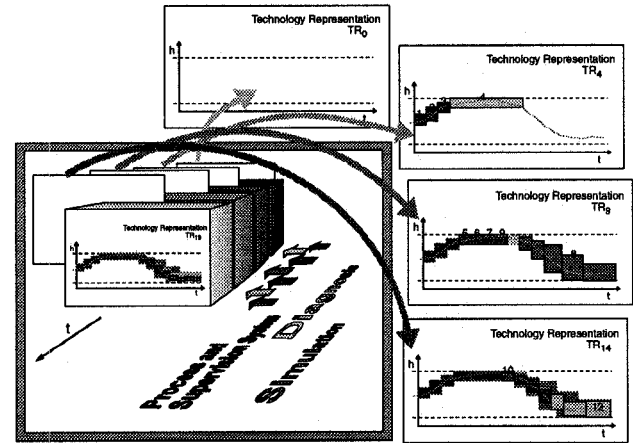


Figure 2: Dynamically Changing Knowledge

Consistent plan generation is based on checking constraints with respect to some underlying knowledge. As introduced above, the crucial deductive task is constraint refutation. Planning starts at some time t_0 over some so-called technology representation TR_0 . As time proceeds during plan generation, the underlying knowledge is changing as well. In fact, there

is a sequence of technology representations containing changing histories of process parameters as depicted below. This dynamics of the knowledge base exhibits the difficulty of designing consistent planning algorithms: For every particular constraint refutation problem, one needs to identify the particular knowledge base it should be based on.

In logical terms, every knowledge base is a certain model. To every time point t , there is related a particular model TR_t . Therapy plan generation beginning at some time point t_0 may be based on

$$TR_{t_0}, TR_{t_0+1}, TR_{t_0+2}, TR_{t_0+3}, \dots$$

Our basic therapy plan generation algorithm (cf. [14]) is exclusively based on the initial model TR_{t_0} . It is one of the benefits drawn from the development of a logic underlying therapy plan generation that it provides a firm basis to clearly distinguish several approaches to consistent plan generation.

As a complete presentation of the investigated algorithmic variants requires some graph-theoretic notions and notations exceeding the space of this short paper, we confine ourselves to a listing of the crucial differences. Variants of plan generation algorithms are specified by the following variants of consistency tests:

- **[Basic Algorithm]** All deductions are based on TR_{t_0} .
- **[Variant 1]** Constraint refutation at time point $t_0 + n$ is based on TR_{t_0+n} .
- **[Variant 2]** Constraint refutation at time point $t_0 + n$ is based on TR_{t_0+n} . Additionally, all preceding deductions are revised with respect to TR_{t_0+n} .
- **[Variant 3]** Deductions are organized as in **Variant 2**, but revision is restricted to only those constraints the refutation of which would dispose of the actual deduction task.

The success of the different variants of planning depends on characteristics of the underlying process, on the actual disturbance, and on the peculiarities of knowledge sources for histories (granularity of models for simulation, monotonicity, ...).

A sequence of models as above is called *monotonic*, exactly if it holds

$$\dots \models TR_{t_0+3} \models TR_{t_0+2} \models TR_{t_0+1} \models TR_{t_0}$$

Usually, technology representations which contain only past and present information reflecting real process states form a monotonic sequence of such models. The figure 2 above is also intended to illustrate monotonicity. In case there are knowledge processing modules (for simulation, e.g.) generating episodes about future process parameter values, monotonicity is a crucial property depending both on peculiarities of the process dynamics and on properties of the forecasting devices.

Under monotonicity, every plan generated by **Variant 1** of the planning algorithm is also consistent with

the initial model TR_{t_0} . In this sense, **Variant 1** implements a finer approach than the basic one. Note that this relation may be not true, if monotonicity can not be assumed. There are similar relationships among the other variants.

In case monotonicity can be assumed, there is no need for checking node expansions for those constraints which have been already refuted earlier. Thus, the logical approach points directly to crucial problems of efficiency to implement real-time planning.

To our knowledge, this perspective is a novelty in plan generation for complex dynamic processes. We have not seen before any collection of logically classified variants of planning algorithms as above.

Acknowledgments

This work has been partially supported by the German Federal Ministry for Research and Technology (BMFT) within the Joint Project (BMFT-Verbundprojekt) WISCON on *Knowledge-Based Process Supervision and Control* under contract no. 413-4001-01 IW 206. The workshop participation of the first author has been supported by the German Research Fund (DFG) under grant 477/378/96. Our work presented here has drawn some benefit from lots of heated and fruitful discussions with our students Jörg Herrmann, Daniel Kirsten, Torsten Lehmann, Daniel Matuschek und Christoph Zschiesche on several aspects of knowledge-based process supervision and control. In using the logic approach described here in their own work, they provided a continuous feedback.

References

- [1] James F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123-154, 1984.
- [2] Oksana Arnold. Reaktive Therapieplanung in dynamischen Prozeßumgebungen. WISCON Report 03/92, HTWK Leipzig (FH), Fachbereich IMN, October 1992.
- [3] Oksana Arnold. Entwicklungswerkzeug zur Analyse generierter Pläne "Plan-Tracer". WISCON Report 03/93, HTWK Leipzig (FH), Fachbereich IMN, December 1993.
- [4] Oksana Arnold. A logic of constraints for dynamic process control. WISCON Report 09/94, HTWK Leipzig (FH), Fachbereich IMN, December 1994.
- [5] Oksana Arnold. Towards structure and management of knowledge bases for controlling. In Eberhard Köhler, editor, *39. Internationales Wissenschaftliches Kolloquium der TU Ilmenau, Band 3*, pages 30-36. Technische Universität Ilmenau, 1994.
- [6] Oksana Arnold and Klaus P. Jantke. Therapy plan generation as program synthesis. In Setsuo Arikawa and K.P. Jantke, editors, *Algorithmic Learning Theory, AIL'94 & ALT'94*, volume 872 of *LNAI*, pages 40-55. Springer-Verlag, 1994.

- [7] Oksana Arnold and Klaus P. Jantke. Therapy plan generation in complex dynamic environments. ICSI Report TR-94-054, International Computer Science Institute, Berkeley, California, October 1994.
- [8] Oksana Arnold and Klaus P. Jantke. Therapy plans as hierarchically structured graphs. WISCON Report 02/94, HTWK Leipzig (FH), FB Informatik, Mathematik & Naturwissenschaften, April 1994.
- [9] Oksana Arnold and Klaus P. Jantke. Therapy plans as hierarchically structured graphs. In *Fifth International Workshop on Graph Grammars and their Application to Computer Science*, Williamsburg, Virginia, USA, November 1994.
- [10] Dietrich Balzer, Volkmar Kirbach, and Volker May. Knowledge based process control. In Hartwig Steusloff and Martin Polke, editors, *Integration of Design, Implementation and Application on Measurement, Automation and Control*, INTERCAMA Congress 1992, pages 33-49. Oldenbourg, 1992.
- [11] Richard E. Fikes and Nils J. Nilsson. STRIPS: A new approach to theorem proving in problem solving. *Artificial Intelligence*, 2:189-208, 1971.
- [12] Klaus P. Jantke and Oksana Arnold. Graph-grammatik-Konzepte in der Therapieplanung für komplexe dynamische Prozesse. In Markus Holzer, editor, *4. GI Theorietag "Automaten und Formale Sprachen"*, 1994, *Proc., Herrsching*, pages 42-47. Universität Tübingen, Wilhelm-Schickard-Institut, 1995.
- [13] Klaus P. Jantke and Oksana Arnold. Inductive program synthesis for therapy plan generation. Communications of the Algorithmic Learning Group CALG-01/95, Hochschule für Technik, Wirtschaft und Kultur Leipzig, FB Informatik, Mathematik und Naturwissenschaften, July 1995.
- [14] Klaus P. Jantke and Oksana Arnold. Variants of plan generation for complex dynamic systems. In Xin Rao, editor, *Proc. 8th Australian Conference on Artificial Intelligence (AI'95)*, November 13-17, 1995, Canberra, Australia, pages 531-538. World Scientific Publ. Co., 1995.
- [15] Klaus P. Jantke and Oksana Arnold. Inductive program synthesis for therapy plan generation. *New Generation Computing*, 14(4), 1996.
- [16] Klaus P. Jantke, Oksana Arnold, and Torsten Lehmann. Flexible simulation scenarios for real-time planning in dynamic environments. In *Proc. Florida AI Research Symposium (FLAIRS-96), Special Track on Real-Time Planning and Reacting*, Key West, FL, USA, May 20.-22., 1996, 1996.
- [17] Klaus P. Jantke, Torsten Lehmann, and Oksana Arnold. Meeting real-time constraints in therapy plan generation for complex dynamic systems by flexible simulation scenarios. In *Proc. ECHTZEIT'96, Karlsruhe, June 18.-20., 1996*. Franzis-Verlag, 1996.
- [18] Klaus P. Jantke and Daniel Matuschek. Developing and implementing planning heuristics in Prolog. In *Proc. Florida AI Research Symposium (FLAIRS-96)*, Key West, FL, USA, May 20.-22., 1996, 1996.
- [19] Daniel Kirsten. Properties of formal languages of therapy plans created by graph grammars. Communications of the Algorithmic Learning Group CALG-04/95, Hochschule für Technik, Wirtschaft und Kultur Leipzig, FB Informatik, Mathematik und Naturwissenschaften, September 1995.
- [20] Daniel Kirsten. Redundant edges in therapy planning. Communications of the Algorithmic Learning Group CALG-06/95, Hochschule für Technik, Wirtschaft und Kultur Leipzig, FB Informatik, Mathematik und Naturwissenschaften, December 1995.
- [21] Daniel Matuschek. PIG: Therapieplanung mit Mitteln der logischen Programmierung. Communications of the Algorithmic Learning Group CALG-02/95, Hochschule für Technik, Wirtschaft und Kultur Leipzig, FB Informatik, Mathematik und Naturwissenschaften, July 1995.
- [22] Daniel Matuschek, Klaus P. Jantke, and Oksana Arnold. Generierung von Therapieplänen mit Mitteln der logischen Programmierung. In Peter H. Schmitt, editor, *Logik in der Informatik, 3. Jahrestagung der GI-Fachgruppe 0.1.6*, pages 75-78. Universität Karlsruhe, Fakultät für Informatik, Bericht 23/95, Juni 1995.
- [23] Volker May, Oksana Arnold, and Uwe Metzner. Wissensverarbeitung in dynamischen Prozeßumgebungen - Eine Anforderungsspezifikation. WISCON Report 01/92, HTWK Leipzig (FH), Fachbereich IMN, March 1992.
- [24] Yoav Shoham. *Reasoning about Change*. MIT Press, Cambridge, MA, 1988.
- [25] Brian C. Williams. Doing time: Putting qualitative reasoning on firmer ground. In *5th National Conference on Artificial Intelligence, AAAI-86, Philadelphia, Pennsylvania, USA, 1986*, pages 105-112, 1986.
- [26] Christoph Zschiesche. Erstellung und Analyse von Wissensbasen der Prozeßsicherung und -optimierung zur Verbesserung und Erweiterung geeigneter Wissensrepräsentationen. Master's thesis, Leipzig University of Technology, August 1995.