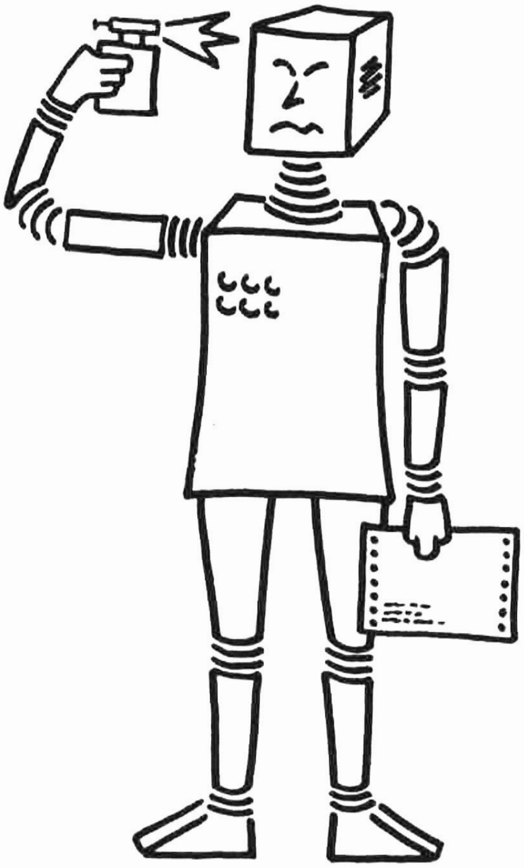
SEKI-PROJEKT

SEKI MEMO

Fachbereich Informatik
Universität Kaiserslautern
Postfach 3049
D-6750 Kaiserslautern 1, W. Germany



Representing and Analyzing

Time and Causality
in HIQUAL Models

Hans Voss

MEMO SEKI-85-07

REPRESENTING AND ANALYZING TIME AND CAUSALITY IN HIQUAL MODELS

MEMO SEKI-85-07

Hans Voss
Universitaet Kaiserslautern
Fachbereich Informatik
Postfach 3049
D-6750 Kaiserslautern
West Germany

Representing and Analyzing Time and Causality in HIQUAL Models

Hans Voss

Abstract: Interest in fertile representations of complex mechanisms is widely increasing. For a variety of reasons many present AI-systems either anticipate or already have a desperate need for a deeper understanding of their respective domains of discourse.

HIQUAL is both a representation language and a tool for the analysis of the structure and behavior of complex systems. Individual components may be arranged horizontally as a set of highly modular, communicating model instantiations, designed to represent one level of abstraction. Separated from the individual models, vertical relations between models at different levels of abstraction allow for the formation of hierarchical representations. This paper focuses on the temporal and causal analysis of HIQUAL models. The presentation is mainly based on a simple but detailed example.

1 Introduction

A common source of many significant problems encountered in the field of expert systems and other areas of R&D in AI is the lack of both common sense world knowledge and flexible knowledge about the structures and possible behaviors in the domains of discourse. In accordance with [Hart 82] and similar discussions in [Chandrasekaran 83], this latter kind of knowledge will be referred to as **deep knowledge**.

1.1 Overview on HIQUAL

HIQUAL has been designed and - by now, that is August '85 - partially implemented as a representation system for hierarchical and qualitative knowledge, mostly about technical systems. Easy development and analysis, versatile usage, and comfortable management of deep knowledge structures are its major design goals. The achievements and mechanisms of HIQUAL shall now be surveyed with respect to these goals.

Easy Development: Composite aggregates reflecting the structure of the system to be modeled may be constructed from individual modular components. These components in turn may be newly constructed for this application, or they may be fetched from a library of already existing system components. Of course, components of one aggregate may themselves be aggregates comprising many other objects.

Easy Analysis: While it should be possible to model a system as concisely as needed, the complexity of the representation should not prohibit a detailed analysis of the possible behaviors of the system. In HIQUAL, the general behavior of each component prototype is first analyzed in isolation, before all components of one level combine their behavior descriptions producing a representation of the overall behavior. This complexity reduction, too, is a consequence of the modularity of the individual components.

Versatile Usage: In order to support many diverse application processes, deep knowledge should be independent from the respective application. In other words: different problem solving strategies like forward/backward chaining or "mixed" control, supplemented by appropriate knowledge for the considered problem class (e.g. explanatory, diagnostic, design or planning knowledge) should all rely on the same kernel of deep knowledge.

For the time being, HIQUAL has not been field tested in a realistic setting. Therefore, hard facts must be substituted by a fair optimism w.r.t. this goal.

Comfortable Management: Supporting efficient development cycles, individual objects as well as (sub-) system hierarchies of general interest may be stored in libraries. Modification of objects for special uses is facilitated by automatic considerations of many consistency checks among the objects in the hierarchies. New hierarchies may be constructed top-down, bottom-up, or in a free mix of both techniques.

This paper only discusses the temporal and causal analysis of a system of HIQUAL models. Representing hierarchies of models is an additional subject of a forthcoming report.

1.2 Related Work

The current shape of the HIQUAL language and the HIQUAL analysis system has been influenced by at least three concepts developed in rather diverse fields.

1. Although the semantic structures differ in many details, the overall syntactic structure and the principal asynchronous behavior of different HIQUAL models is similar to agents in the actor language CSSA (e.g. [Beilken/Mattern/Spenke 82], the development of which the author has participated in.

2. The basic terminology for the temporal analysis of individual objects and compound systems is based on the representation of temporal intervals in [Allen 83] and [Vilain 82]. Similar techniques for representing temporal relations between events in the physical world have been applied in the KIT project in Berlin, where an event is regarded as the meaning of part of a natural language text [Guenther et al 83].

3. The general structure of the HIQUAL representation and analysis system in some respects resembles the ENVISIONING system (e.g. [de Kleer,Brown 83,84]). Unlike ENVISIONING, HIQUAL does not base its behavioral analysis on qualitative differential equations as descriptions of the naive physics of the domain. Compared with ENVISIONING, the modeling approach suggested by HIQUAL is not connected as closely to the models of concrete physics. However, it provides more elaborate facilities for the specification and the automatic synthesis of qualitative temporal relations between events and the states of the objects.

The independent development of QRL [Raulefs 84] intensively describes the behaviors of objects and processes by explicitly specified structures of temporal intervals. In HIQUAL, most of the corresponding structures would be derived automatically during the temporal analysis without the need for explicit representations.

1 Introduction

As a general reference to other related work, citing the 'Special Volume on Qualitative Reasoning about Physical Systems' of the journal 'Artificial Intelligence' [AI Journal 84] should suffice.

2 Qualitative Relations Between Temporal Intervals

The basic mechanism for the representation of temporal knowledge in HIQUAL models and in HIQUAL's temporal analysis (TA) is defined in [Allen 83,84]. Therefore, only the main characteristics are surveyed:

1. For technical reasons and in substance, there exist only **non-zero temporal intervals.** Nevertheless, TA incorporates an extended mechanism for designating special intervals as **qualitative time points** (cf. section 4.1).

2. Between two concrete intervals X and Y, exactly one of **seven disjoint relations or one of their inverses** may hold (see Fig 1). In the following, let RALLEN be the set of all 13 possible relation names.

3. **Uncertainty or missing knowledge** about the concrete relation between two intervals can be represented by stating all relations which might hold. E.g. for intervals I1 and I2, and R⊆RALLEN, I1 -- R --> I2 is interpreted as follows: if I1 -- r --> I2 holds for some r∈RALLEN, then r∈R.

4. Given some relations between a subset of intervals, the resulting reduced relations between all intervals can be computed by **constraint propagation**, using a complete transitivity matrix for each combination of two of the 13 relations.

Example: I1 -- starts-in --> I2 ∧ I2 -- m --> I3
⇒ I1 -- starts-before --> I3

where starts-before := {<,m,o,fi,di},
starts-in := {=,d,oi,s,si,f}.

Let R⊆RALLEN; then

- inv(R)⊆RALLEN is the result of substituting every relation in R by its inverse relation,

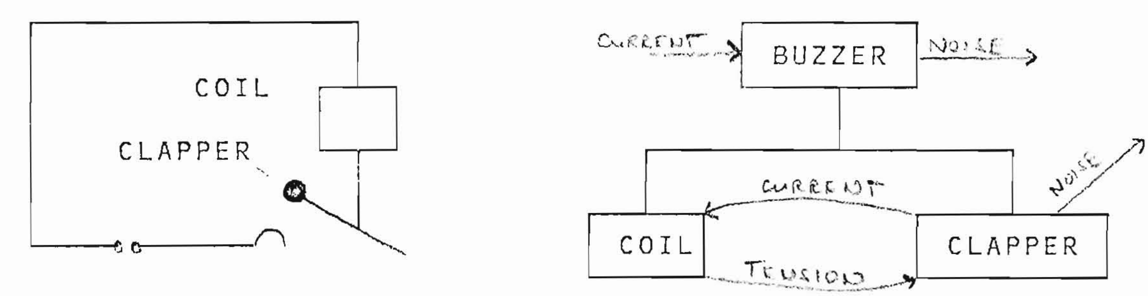
- not(R) := RALLEN\R is the complement of R.

Example: - I1 -- <,m,o,fi,di --> I2 ⇔
I1 <-- inv(<,m,o,fi,di) -- I2 ⇔
I1 <-- >,mi,oi,f,d -- I2

- I1 -- not(starts-before) --> I2
⇔ I1 -- =,>,mi,oi,f,d,s,si --> I2

relation	symbol	inverse	graphics
X before Y	<	>	xxx yyy
X equal Y	=	=	xxx yyy
X meets Y	m	mi	xxxyyy
X overlaps Y	o	oi	xxx yyy
X during Y	d	di	xxx yyyyy
X starts Y	s	si	xxx yyyyy
X finishes Y	f	fi	xxx yyyyy

Fig.1: The temporal relations of Allen.
(Fig. taken from [Allen 83])



(2a) (2b)

Fig.2a: A naive picture of a buzzer
Fig.2b: A two-level buzzer hierarchy

3 Defining Models in HIQUAL

An example being adopted from [de Kleer/Brown 83] describes the behavior of a **buzzer** with its two components: a coil and a clapper. Fig. 2a shows a naive picture of the buzzer structure. A two-level hierarchy of the buzzer is depicted in fig. 2b.

The HIQUAL model definitions for COIL and CLAPPER (fig. 3 and 4) will be first explained informally. The following presentation of the result of the temporal analysis in essence can be viewed as a formal definition of the semantics of the models.

A model of type CLAPPER has an input port TENSION-I of type FORCE and two output ports CUR-O and NOISE. NOISE of type SOUND has three possible values: usually, there is no noise (quiet); and only for instants of time, the noise may rise to a low or high value, resp. (The is-inst-construct is used here merely for demonstration without the objective of realistic modeling). In the development of large systems, commonly used type and function definitions (functions are not needed in this example) may be stored to and fetched from global databases [Scherer 85].

A CLAPPER model is either in facet CLOSED or OPENED. Both facets are termed as **okfacets**. The distinction between okfacets and **exfacets** (exception facets) as their counterparts is only relevant for special application processes such as diagnosis or monitoring. Exception facets are not further needed and discussed in this paper. Each facet consists of only one rule. Facet CLOSED is left when a value **high** is received in port TENSION-I. For later reference in the relations-clause the precondition TENSION-I=high has been provided with the label 'a'. The comma between the two actions of CL indicates that the syntactic ordering of the actions in the right hand side (rhs) of a rule is semantically not important. The effect of rule CL is a switch to the new state OPENED, and issuing a 'CUR-O=0' - message to the model connected with port CUR-O.

Rule OP of CLAPPER and the definition of COIL work according to the same scheme. In all cases, the interpretations of the relations-clauses will be explained later. Finishing the informal explanation, each instantiation of CLAPPER (COIL) is restricted to be initially in facet OPENED (OFF).

```
type CLAPPER is model

  types CURRENT is (neg;0;pos);
        FORCE   is (low;high);
        SOUND   is (quiet; low is inst; high is inst);

  ports in  TENSION-I is FORCE;
        out NOISE is SOUND,
            CUR-O is CURRENT;

  okfacet CLOSED is
    CL is preconds (TENSION-I=high a);
          actions  enter OPENED, CUR-O=0;
          relations facet is im with a;

  okfacet OPENED is
    OP is preconds (TENSION-I=low a);
          actions  enter CLOSED, (NOISE>quiet b),
                                 (CUR-O>0 c);
          relations a is fi,o,di with facet,
                    b is im with facet,
                    c is im with facet;
  initially OPENED;
end CLAPPER;
```

Fig.3: The HIQUAL definition of CLAPPER

```
type COIL is model

  types CURRENT is (neg;0;pos);
        FORCE   is (low;high);

  ports in  CUR-I is CURRENT;
        out TENSION-O is FORCE;

  okfacet ON is
    ON-R  is preconds (CUR-I=0 a);
             actions  enter OFF, (TENSION-O=low  b);
             relations b is del with a;

  okfacet OFF is
    OFF-R is preconds (CUR-I>0 c);
             actions  enter ON, TENSION-O=high;
             relations c is fi,o,di with facet;
  initially OFF
end COIL
```

Fig.4: The HIQUAL definition of COIL

4 The Temporal Analysis of Single Models

Before analyzing a system of models, each individual model of the system will be examined in separation.

4.1 Associating Temporal Intervals with States and Events

TA defines the semantic of HIQUAL models by determining the possible behaviors of the models. Only some of the important concepts can be illustrated in this paper.

Behavior is essentially described as a set of relations between the temporal intervals associated with the events and states of the model executions. In the ontology of TA, **states** are instantiations of facets. An **event** is defined as the "occurence" of

- an action on the rhs of a rule, which is not an enter-action,
- a matching condition on the lhs of a rule.

States and events are partially denoted by the syntactical objects (conditions, actions, facets) of the model. Unique denotations for states and events, and hence, for the intervals in which they are occuring, are obtained by adding the following qualifications:

- the name of a model instance, or the name of the model type if only one instantiation is considered;
- the name of the rule in which the condition or action occurs (not needed for states);
- a number, say i, for the i-th instance of the state/condition/action during the (hypothetical) model execution. i is called the **index** of the interval.

In addition to these parameters, a fourth parameter specifies the type of the occurence as one of the values **in** (instantaneous), **ni** (not instantaneous). The type of an occurence is 'in', if its respective facet or value declaration in the model includes the 'is-inst'-clause. Unfortunately, there is insufficient space to go into the exact details of the consequences of an 'is-inst' specification. Principally, the possible temporal relations, say R', of an instantaneous occurence o_1 with another occurence o_2 is restricted to a subset of the relation set, say R, which would be established if o_1 were non-instantaneous. The concrete shape of the restriction depends on the type of o_2:

- if o_2 is non-instantaneous, then R' := R\{o,=,fi,si,oi,di}; i.e. o_1 must not be equal with o_2 or have both a common part and a disjoint part with o_2.

- if o_2 is instantaneous, then R' := R\{o,s,d,f,fi,si,oi,di}; i.e. two instantaneous events either are equal, or they meet, or one is before the other.

Example: Let inst1, inst2 be instantaneous events;
ninst be a non-instantaneous event;
inst1 -- m --> ninst <-- m -- inst2;

then inst1 -- = --> inst2 is the result of the extended propagation algorithm respecting the types of intervals.

Allen's propagation algorithm would establish inst1 -- f,fi,= --> inst2. In contrast to Allen, and similar to our approach, Vilain expands the logic of intervals to include time points. However, there are two main differences:

- Vilain does not allow for a meets-relation (m) between a time point and another time point or an interval. (Hence, the example from above would not work because its preconditions cannot be satisfied.) This seems reasonable because Vilain's time points really correspond to points - say, on the real axis - whereas instantaneous states/events in HIQUAL are still considered as intervals, if only very small ones.

- For each possible relation between two time points, and between a time point and an interval, Vilain introduces a new temporal relation. Conceptually, allowing the 'old' relations to respect the types of their arguments, and only manipulating the result of the propagation operation seems to be the simpler idea.

TA does not really execute the models, but tries to establish the set of all possible behaviors. Therefore, TA usually refers to fixed but undetermined temporal intervals by using variable indices or more general "index expressions". In unambiguous contexts, the instantiation and the rule parameter are optional. Let <caf> be a condition, an action, or a facet; <inst-name> be an instantiation name (or a model type name); type ∈ {ni,in}; then

<caf> [/ [<inst-name>] / [<rule name>]] / <type> / <index>

is the general format of event/state-expressions, **ES-expressions** for short. Arguments enclosed in square brackets are optional. For example, OFF/COIL/ON-R/ni/1; NOISE>quiet//OP/in/j; CUR-I=0/ni/i-1

are valid ES-expressions.

4.2 Establishing Temporal Relations

The general format of a temporal relation between two ES-expressions is an extension of Allen's notation:

<es$_1$> —— [<ont-type> /] <arel> ——> <es$_2$> (*)

where <ont-type> is the "ontological" type of the relation (see below); <arel> is a subset of RALLEN; <es$_i$>,$i \in \{1,2\}$, are ES-expressions. Existing ontological qualifications must be explicitly specified, otherwise it will be assumed that no such qualifications exist.

The essential result of TA is a **set** of these relations. In a rather crude presentation of the formal background, an interpretation Γ of this set associates a concrete interval Γ(es) with each ES-expression es in consideration, such that for every temporal relation specification (*) the following holds:

Γ(<es$_1$>) -- <arel> --> Γ(<es$_2$>)

Please notice the dashed arrow (cf. chapter 2) being used for the interpreted expressions. A more formal treatment must take some care for generating consistent substitutions of multiple occurences of index variables.

4.2.1 Ontological Relationships

The following ontological qualifications are distinguished by TA:

- if a new state is established in an enter-action of a rule R, then (the ES-expression for) this state **is defined by** (id) the events denoted by the conditions of R; the 'old' state corresponding to the facet encompassing R **is left by** (il) these conditions.

- events denoted by non-enter-actions on the rhs of a rule are said

  - to **be effected in** (ie) the state in which they are occuring. This state is either denoted by the facet name of the enter-action of R, or, if no enter-action exists, by the facet R belongs to.

- to **be caused by** (ic) events denoted by the conditions on the lhs of R.

- an event denoted by a condition in a rule R without enter-action **is registered** (ir) in the state corresponding to the facet in which R is defined.

If a relation containing 'id','ie','ic','il', or 'ir' is inverted (building **inv** of the temporal relation), then the ontological qualification will also be inverted to 'd', 'e', 'c', 'l', or 'r', resp.

TA associates a **default temporal relation** with each ontological relationship:

- the temporal aspects of id and ic are defined by the standard relation not(starts-before). Effects may not start before their causes.

- the temporal aspect of ie is represented by the relation starts-in.

- 'il' is associated with the relation not(<,mi).

- 'ir' is associated with the relation not(>,m). [Notice that inv(not(>,m)) = not(<,mi), which is the temporal relation for il. Indeed, il can be considered as a special case of 'r' (registers).]

These standard relations allow for very liberal interval relations, i.e. only the start points of the intervals are restricted, whereas the end points are totally undetermined. Hence, delayed effects as well as causes and effects occuring simultaneously (like e.g. in a system of connected gears) may be modeled. In practice, however, one often knows much more about the relative extensions of the states and events to be modeled. For these cases, HIQUAL provides facilities for specifying **subsets** of the standard relations. Some special care has to be taken that the possible relations are even more restricted if instantaneous events or states are part of the relation. The definitions of COIL and CLAPPER in fig. 3 and 4 offer some examples making use of these facilities by means of the temporal relations-clause. The effects of these restrictions can be examined in fig. 5 and 6. which depict the result of TA for the individual models of CLAPPER and COIL. For better reference in the text, the relations occuring in these figures have been labeled. Here are some examples:

- the condition 'a {TENSION-I=low} is (fi,o,di) with facet' in rule OP leads to relation CL4 (fig.5). Notice that CL4 uses inv(fi,o,di)=(f,oi,d);

- 'c {CUR-O>O} is im with facet' in rule OP generates the restricted relation CL6. 'im' is a shorthand for (=,s,si), expressing the **immediate** start of the state w.r.t. the event.

- 'b {TENSION-O=low} is del with a {CUR-I=O}' in ON-R yields CO7 (fig.6). 'del' as a shorthand for (f,oi,d) expresses the **delayed** effect of the tension going down to 'low' due to zero current.

Notice that the explicit restriction of more than one standard relation may lead to inconsistencies recognized by generating an empty relation during the constraint propagation. Unfortunately, the development of all relations occuring in the figures cannot be explained in this paper.

4.2.2 The Continuity Assumption

Most models of the physical world make the general assumption of **continuity**, i.e. they work according to the (simplifying) principle of variables changing their values continuously. HIQUAL respects the continuity assumption by controlling the value transitions in in-ports for possible departures from continuity. The current implementation only allows for explicit continuous transitions. As an example, consider the following hypothetical situation: for a port P of type T=(t1;t2;t3) TA recognizes that P changes its value from t1 to t3. TA will indicate this situation as a potential flaw in the modeling. In a more advanced implementation, TA will probably add to its result the explicit information that P must have had the value t2 during an interval immediately between the intervals for the values t1 and t3.

The incorporation of the continuity assumption is one of the important differences between TA and "usual" temporal semantics of distributed systems.

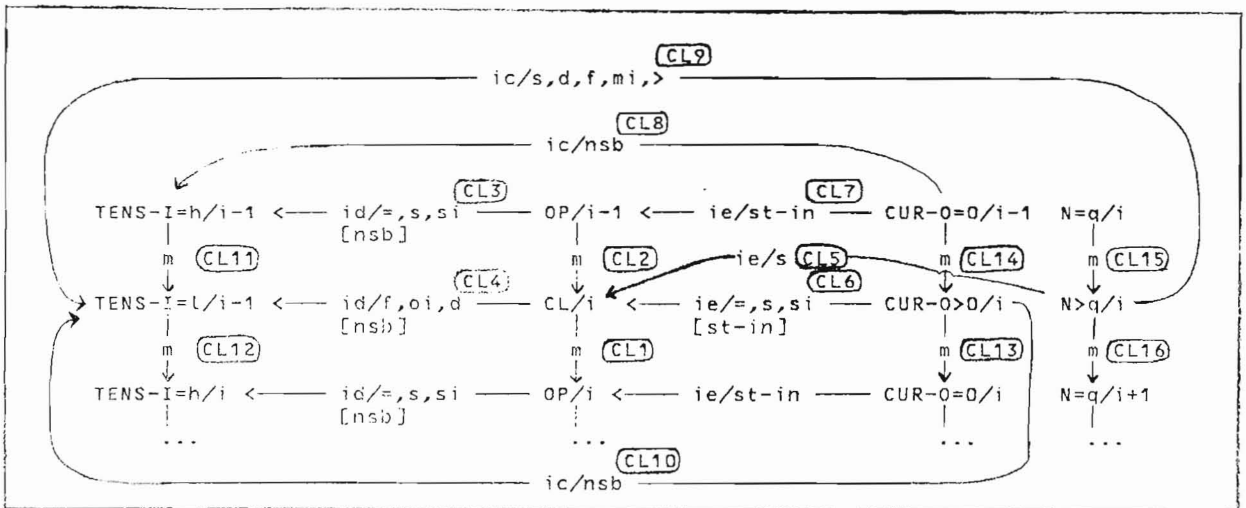


Fig.5: Temporal relations for CLAPPER.
(The figure uses obvious abbreviations for the ES-expressions of CLAPPER. Standard relations being overwritten by explicit restrictions are shown in square brackets.
nsb := not(starts-before); st-in := starts-in)

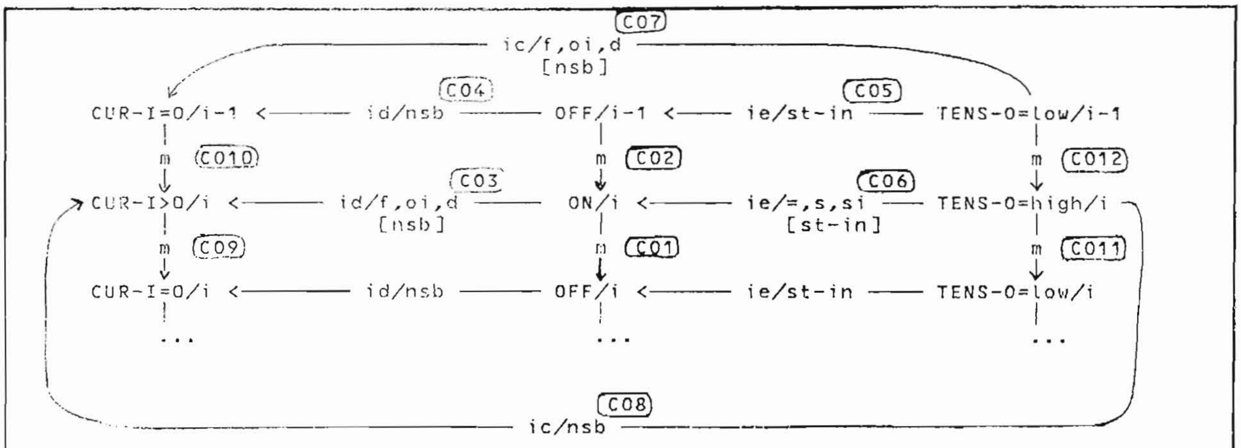


Fig.6: Temporal relations of COIL.
(For further remarks, cf. fig.5)

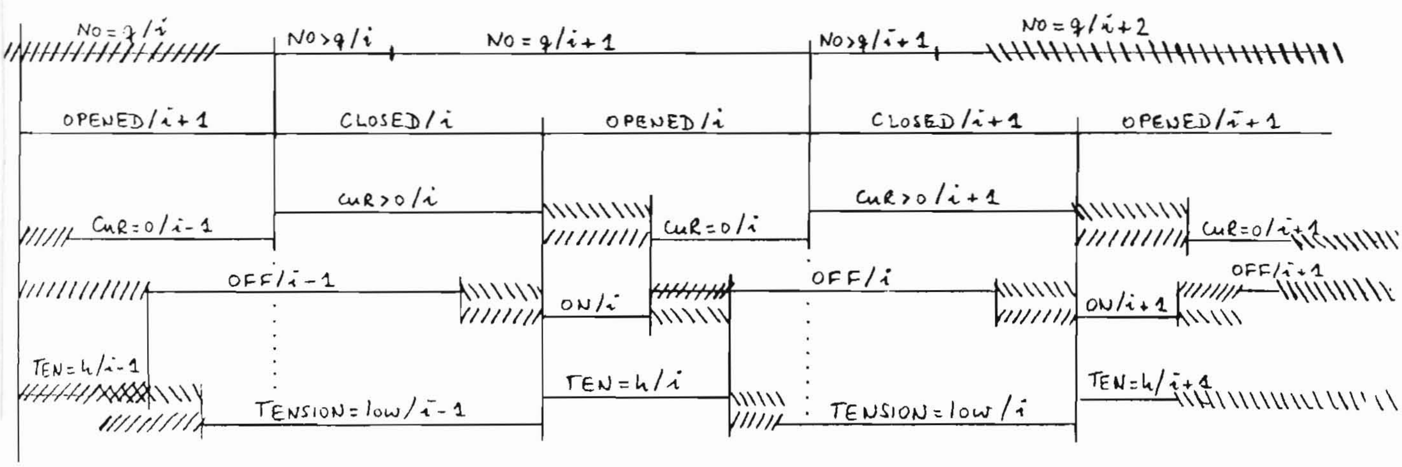


Fig.7: Interval graphic for the CLAPPER/COIL system

5 Analysis of a System of Models

A system of models is defined by instantiating and connecting its individual components. E.g., a CLAPPER/COIL system can be defined by

```
level BUZZER-1 is
  models C is COIL, CL is CLAPPER;
  connect CUR-I(C) = CUR-O(CL),
          TENSION-I(CL) = TENSION-O(C);
```

The important effect for TA consists of the "temporal identifications" of the in- and out-ports, and hence the possible propagation of temporal relations across the borders of the individual models. In the simplest possible case, an out-event of a sender may be exactly identified with a corresponding in-event of its receiver. In the example, the combination of the two models can be represented by adding the following relations to the union of the CLAPPER and COIL relations:

```
COMB1:  CUR-I=0/ni/i ── = ──> CUR-O=0/ni/i
COMB2:  CUR-I>0/ni/i ── = ──> CUR-O>0/ni/i
COMB3:  TENSION-I=low/ni/i ── = ──> TENSION-O=low/ni/i
COMB4:  TENSION-I=high/ni/i ── = ──> TENSION-O=high/ni/i
```

In general, these combination relations are not as simple as in the example. If, for example, COIL would receive 'CUR-I=1' instead of 'CUR-I>0' in OFF-R (a suitable definition of CURRENT presupposed), while CLAPPER is still sending "only" 'CUR-O>0', then many continuations are possible, depending on whether CUR-O increases up to the value 1 (and if, how often ?) or remains below.

Presenting the result of TA as the effect of propagating COMB1-COMB4 as a new set of linearly listed relations would be boring both for the reader and the author. Instead, fig.7 visualizes the final relation set by an interval graphic depicting two cycles of the BUZZER-1 system. The labeling of the intervals deliberately uses obvious abbreviations. A "slashed" part of an interval covers its possible starting points, a "back-slashed" part of an interval denotes its possible end points.

There are many interesting observations the reader is invited to make on her or his own. As a start, one might be attentive to the different overlapping relations between OFF and CLOSED on the one hand, and ON and OPENED on the other hand. One thing the graphic shows at any rate, namely that even in such simple examples the combination of individually simple components may exhibit sophisticated behavioral details which scarcely can all be recognized in advance.

6 Conclusions

HIQUAL has been described as a representation language and as a system for the qualitative temporal and causal analysis of (physical) systems. The results of the local analysis of each individual model can be merged to obtain the global behavior of a whole system of connected models. HIQUAL is distinguished by integrating such concepts as

- (systems of) highly **modular models** constructed according to the message passing paradigm;
- the interpretation of states **and** events as temporal intervals denoted by suitably indexing the corresponding syntactical elements in the model definitions;
- the interpretation of messages as **continuous** flow of material and forces;
- the permission of instantaneous events and states as specially handled temporal intervals;
- the building of **hierarchies** of object representations (not discussed in this paper).

7 Future Work

Much work remains to be done both at the conceptual and the implementation level.

An implementation of a precursor of HIQUAL including a data base system for the management of models, model hierarchies, and global type and function definitions to be used in different models has been completed in spring '85 on a Symbolics Lispmachine. This system has been tested by a three level modeling of an internal-combustion engine (without a complete temporal analysis as discussed in this paper) [Scherer 85]. The experiences with this first prototype led to the integration of qualitative temporal relations, as discussed in this paper. Implementation of the new system is in progress.

The conceptual work will be extended in several directions, where the focus will be on

- the refinement of the data flow analysis for HIQUAL models, i.e. for generating more accurate index informations;
- the more profound elaboration of the underlying temporal logic;
- refinement of the temporal specifications including duration reasoning;
- extensions of the basic framework in the direction of specific application classes like explanation and justification components being used by "conventional" expert systems. This will certainly include the formation of processes and episodes (cf. [Forbus 84]) as suitable abstractions for the behaviors established by TA.

Acknowledgements: This work has been influenced by many discussions with Michael Th. Reinfrank, Werner Scherer, Marc Linster, and Hans-Werner Eiden. The presentation profits from helpful suggestions by Vijay Bandekar, Werner Dilger, and Peggy Johnson. There has been a situation where only the immediate and willing assistance of Joerg Siekmann guaranteed the continuation of this work.

8 Literature

AI-Journal 84 : Artificial Intelligence (1984), Vol.24, Numbers 1-3, Special Volume on Qualitative Reasoning about Physical Systems.

Allen, James F. (1983) : Maintaining Knowledge about Temporal Intervals, CACM, Vol.26, No.11, 832-843.

Allen, James F. (1984) : Towards a General Theory of Action and Time, Artificial Intelligence, Vol.23, 123-154.

Beilken, Christian; Mattern, Friedemann; Spenke, Michael (1982) : Entwurf und Implementierung von CSSA, Memo SEKI-82-03 (6 volumes), Fachbereich Informatik, Universitaet Kaiserslautern.

Chandrasekaran, B. (1983) : Towards a Taxonomy of Problem Solving, AI Magazine, Vol.IV, No.1, 9-17.

de Kleer, Johan; Brown, John S. (1983) : Assumptions and Ambiguities in Mechanistic Mental Models, in D. Gentner/ A. L. Stevens (Eds.): Mental Models, Lawrence Erlbaum Associates, 155-190.

de Kleer, Johan; Brown, John S. (1984) : A Qualitative Physics Based on Confluences, in [AI Journal 84], 7-83.

Forbus, Kenneth D. (1984) : Qualitative Process Theory, in [AI Journal 84], 85-168.

Guenther, Siegfried; Habel, Christopher; Rollinger, Claus-Rainer (1983): Ereignisnetze : Zeitnetze und referentielle Netze, KIT Report 12, Fachbereich Informatik, Technische Universitaet Berlin.

Hart, P.E. (1982) : Direction for AI in the Eighties, SIGART Newsletter, 79:11-16.

Raulefs, Peter (1984) : Foundation of Expert Systems for Conceptional Design in Mechanical Engineering, Memo SEKI-84-08, Fachbereich Informatik, Universitaet Kaiserslautern.

Scherer, Werner (1985) : Ein Repraesentationssystem fuer hierarchisch strukturiertes Tiefenwissen, Diplomarbeit, Fachbereich Informatik, Universitaet Kaiserslautern.

Steele, Guy Lewis Jr. (1980) : The Definition and Implementation of a Computer Programming Language Based on Constraints, AI-TR-595, MIT AI-LAB.

Vilain, Marc B. (1982) : A System for Reasoning about Time, Proceedings of AAAI-82, National Conference on Artificial Intelligence, 197-201.