A Fast History-oriented Envisioning Method Introducing Temporal Logic

Takashi Washio
Research Center for Safety Science Mitsubishi Research Institute, Inc.
ARCO TOWER 1-8-1 Shimomeguro Meguro-ku Tokyo 153 Japan
phone +81-3-5434-8955 fax +81-3-5434-8951
e-mail: washio@mri.co.jp

Masaharu Kitamura
Nuclear Engineering Department Tohoku University
Aramaki Aza Aoba,Aoba-ku, Sendai 980, Japan

Abstract: The history-oriented envisioning (HOE) we have already proposed is a novel and generic envisioning method focusing on our intentional behaviors and actions. A knowledge representation named as "partial history" has been introduced to partially specify behaviors and actions we intend. Based on multiple attainable envisionings, the HOE derives an envisionment satisfying the specifications. A new method presented in this paper is an extended version of the former HOE. It enables the specifications of our intentional behaviors and actions in form of a temporal logic. The introduction of the temporal logic allows us to specify infinite, implicit and abstracted behaviors and actions. Also, the efficiency of the envisioning under practical conditions is increased by introducing a new total envisioning based algorithm. These flexibility and efficiency of this new HOE method are demonstrated through an example to control a steam generator.

1 Introduction

One of the primary tasks of qualitative reasoning is to envision system "behaviors". The conventional framework of the envisioning consists of "attainable envisioning" and "total envisioning" [de Kleer and Brown 1984; Forbus 1984, 1988; Kuipers 1984, 1986]. The basic idea of these methods is to evaluate sound behaviors of a system while maintaining a set of initially given background assumptions without intentionally changing the assumptions at any intermediate time steps. In contrast, Forbus defined an "action" to introduce our intentional replacement of some background assumptions in a system scenario, and established "action-augmented envisioning". This enumerates all possible transitions among situations consisting of quantity states, views, processes and actions [Forbus 1989]. Besides, Drabble extended the notion of the actions to involve the exogenous specification of quantity states and to have qualitative time intervals [Drabble 1993]. However, a difficulty of combinatorial explosion of derived situations has been reported in both of the conventional and intentional envisioning methods, when they are adopted to practical scale applications [Caloud 1987; Forbus 1989; Forbus and Falkenhainer 1990, 1992; Amador et al. 1993].

As an efficient remedy to this difficulty, the authors have proposed a novel and generic envisioning method called as "History-oriented Envisioning (HOE)" [Washio 1994]. It can restrict the scope of the envisioning by the specification of the partial behaviors and actions we are interested in. Many works on simulation, planning, diagnosis and design in the field of qualitative reasoning utilize the envisioning to obtain the information associated with specific system behaviors. We claimed in our previous work that their efficiency can be highly enhanced by introducing the envisioning focused on specific and meaningful behaviors and actions [Washio 1994]. However, the algorithm of the HOE we have proposed still does not have sufficient reasoning speed for its use in the control and diagnosis applications of practical scale systems.

Another issue of the current HOE is the limited flexibility of the knowledge representation called as a "partial history". It specifies the partial behaviors and actions we intend in the envisioning. The former partial history can merely specify the time series of primitive and snapshot facts such as "amount of water in a pot was 1Kg at a time step, and boiling of water occurred at the next step." This does not allow us to directly specify any contextual behaviors such as "amount of water in a pot is 1kg, until a valve is opened." and "boiling of water occurred intermittently." Kuipers proposed a very attractive idea to use a temporal logic named as "Expressive Behavior Tree Logic (EBTL)" to check the behaviors on his QSIM behavior tree [Kuipers 1994a]. His method can proof if a theorem written in the logic holds for any ordinary differential equation consistent with the behavior tree. The introduction of the temporal logic to the partial history is expected to highly extend the variety of specifications of the partial behaviors and actions.

This paper has two objectives. First, we extend the framework of our former HOE method to introduce the temporal logic into the "partial history", i.e., the specification of our intentional behaviors and actions. Second, we change the algorithm of HOE to improve its reasoning speed under some practical conditions. These improvements are independently applicable to enhance the performance of the HOE. The flexibility and efficiency of the new HOE method are evaluated through the example to control a steam generator.

2 HOE Introducing Temporal Logic

2.1 Extended Partial Situation and Partial History

The fundamental structure of temporal behaviors and actions has been discussed in detail in the past works [Hayes 1979; Forbus 1984, 1989; Williams 1984; Dean and McDermott 1987]. Hayes and Forbus defined a sequence of changes of objects in a scenario as a "history". A history consists of "situations". A situation is either an "episode" or an "event". Events always last for an instant, while episodes usually occur over a time interval. A primitive and snapshot fact in an event or an episode such as "amount of water in a pot is 1kg." is called a "token" [Dean and McDermott 1987]. The formal representation of a token in QPT [Forbus 1984] is a proposition of a quantity state, a view, a process, an action, a relation among them and a transition of one of them at a time (or in a time interval). T-operators are used to state that a particular token is true at some time, and M-operators represent the measured value of a quantity at some time. We call a token represented in the QPT as a "QPT-token" here.

Based on these definitions, we have already proposed some important ideas on the history as follows [Washio 1994].

Definition 1: A "partial event" is a set of some tokens involved in an event in a history. A "partial episode" is a set of some tokens involved in an episode in a history.
Definition 2: A "partial situation"* of a history is either a partial event or a partial episode.

In the extended version of the "partial situation", we allow a token to a contextual formula in the "Expressive Behavior Tree Logic (EBTL)" [Kuipers 1994a]. The elementary temporal operators in the EBTL are (and p q), (not p), (next p) and strong-until(p q), where p and q are the QPT-tokens or the temporal operators. Many other temporal operators such as (eventually p) and (almost-everywhere p) can be generated by the boolean combinations and nestings of these elementary temporal operators. A token in the EBTL is a behavior quantifier of either (possibly p) or (necessarily p), where p is also a QPT-token or a temporal operator. A token represented in the EBTL is named as an "EBTL-token" in our work. The details of the syntax of the EBTL can be seen in [Kuipers 1994a], and their semantics is explained later. The semantic definition of each QPT-token p is equivalent to (necessarily (always p)) in terms of EBTL. However, the conventional QPT-tokens are processed in a different way from the EBTL-tokens in the HOE for the efficiency.

The representation of one of the partial situations for an example depicted in fig.1 is represented as follows.

* The terminology "partial slice" has been changed to "partial situation" because of the higher appropriateness in terms of its definition.

Partial Situation Catching-Ball-under-Flame(?time)
Individuals: ball a ball
(possibly flame a flame)
basket a basket
Quantities: (M A[position-of(ball)] ?time)= $(-\infty$,H1)
(eventually (strong-until
(M Ds[position-of(ball)] ?time) = -1
(T Status(Catch-In(ball, basket),
Activated) ?time))) (1)
Views:
Processes: (T Status(Heat-Flow(flame, ball, flame-ball), Inactive) ?time)
Actions: *(necessarily (T Status(Catch-In (ball, basket), Activated) ?time))*

Each slot contains a list of tokens that must hold within this partial situation in "?time". EBTL-tokens are indicated in italics. This partial situation means that the objects of a ball and a basket must always exist, and also a flame may exists. In the mean time, the ball must be bellow the height H1, and it must continuously descends until it is caught in the basket at least once. Also, the heat flow process between the flame and the ball must not be working at any time. Furthermore, the ball must be definitely caught in the basket at some time within this partial situation. Some slots can be left unspecified as the Views slot in this example.

The term "?time" represents the temporal specification of a partial situation, and follows the conditions indicated below with respect to its duration and the limit hypotheses [Washio 1994].

?time is an instant. $\Leftrightarrow$ start(?time)=end(?time),
?time is an interval. $\Leftrightarrow$ start(?time)<end(?time), (2)
The duration of ?time is unspecified. $\Leftrightarrow$
start(?time) $\leq$ end(?time).

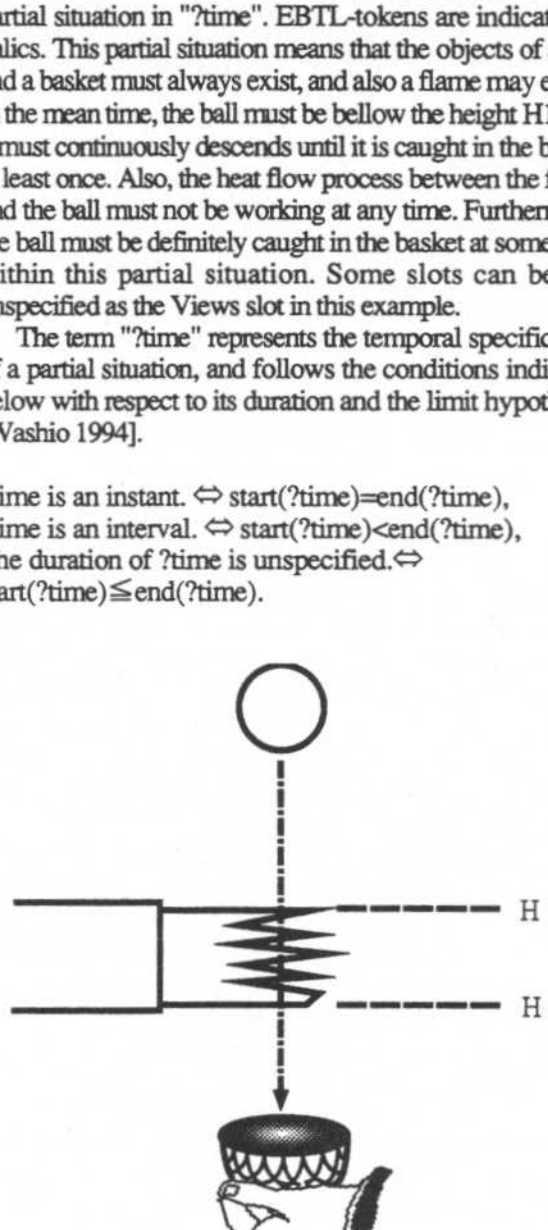


Fig.1 Catching a ball dropped through a flame.

A partial situation involves some limit hypotheses.

$\Rightarrow$?time is an instant. (3)

?time is an interval.

$\Rightarrow$ A partial situation does not involve any limit hypotheses.

The QPT-tokens in the individuals and quantities slots of a partial situation are directly used as the assumptions for the HOE. On the other hand, the QPT-tokens in the views, the processes and the actions slots must be precompiled by unifying them to their domain models in order to obtain their implicit assumptions. The domain models of the views and processes in the QPT show their assumptions of "Individuals", "Preconditions" and "Quantity Conditions" [Forbus 1984]. Also, the domain models of the actions have the ones of "Individuals" [Forbus 1989]. These assumptions are also QPT-tokens. Another important assumption in a partial situation is the duration of "?time". Its specification controls the generation of the limit hypotheses in the process of the HOE. The set of these assumptions of each partial situation k is expressed as Ps_k here. The details of the algorithm to derive Ps_k can be seen in [Washio 1994]. On the other hand, any EBTL-tokens are not involved in Ps_k. The set of EBTL-tokens in a partial situation is expressed as Es_k.

The definition of a "partial history" is given based on the partial situations as described in [Washio 1994].

Definition 3: A "partial history" of a history is a set of partial situations of the history which time intervals and instants are totally ordered in time domain.

A partial history has a list of the T-operators of the QPT to say that a particular partial situation is true at some time. It also involves a list of time constraints on the partial situation. Those constraints follows the rules (2). An example of a partial history for the ball is shown here.

```
Partial History  Initial-and-Final-Ball
  Partial Situations:(T Initial-Position-of-Ball(I0) I0)
                     (T Position-Decreasing-of-
                      Ball-above-Flame(I1) I1)
                     (T Heat-Flow-to-Ball-Active(I2) I2)
                     (T Heat-Flow-to-Ball-Inactive(I3) I3)
                     (T Position-Decreasing-of-
                      Ball-under-Flame(I4) I4)
                     (T Catching-Ball-                  (4)
                      under-Flame(I5) I5)
  Time Constraints:(start(I0)=end(I0)), (end(I0)=start(I1)),
                   (start(I1)<end(I1)),(end(I1)=start(I2)),
                   (start(I2)≦end(I2)),(end(I2)=start(I3)),
                   (start(I3)<end(I3)),(end(I3)=start(I4)),
                   (start(I4)<end(I4)),(end(I4)=start(I5)),
                   (start(I5)=end(I5))
```

This partial history specifies the partial behaviors and actions associated with a ball from its initial position to its final state in a basket.

The HOE utilizes Ps_k of each partial situation k (k=1,...,n) as a temporal part of the background assumptions during the time [strat(?time), end(?time)] in a partial history. Besides the information of EBTL-tokens in the Es_k (k=1,...,n) are used to filter the behaviors in the envisionment.

2.2 EBTL-Filter

Situations and transitions inconsistent with some EBTL-token specified in a partial situation must not be involved in the envisionment. A procedure named "EBTL-filter" is defined to filter consistent situations and behaviors with a given EBTL-token. For the preparation to describe the filter, some basic notions are explained first.

The notations and semantics of EBTL are given in the framework of [Kuipers 1994a]. A behavior tree M is an ordered triple $<S, R, L>$ where S is a set of situations, R is a set of situation transitions, and L is a labeling. The labeling L maps each situation s to an interpretation of all QPT-tokens in s. Given a behavior $x=<s_0,s_1,...,s_n>$ where s_0 and s_n are a starting node and an ending node in the M respectively, for $1 \leqq i \leqq n$ we let x^i denote the behavior $<s_i, s_{i+1},...,s_n>$, which is the subbehavior of x starting at s_i. We write $M,s_0 \models \phi$ (respectively $M,x \models \phi$) to mean that a formula ϕ is true at the situation s_0 (respectively of the behavior x) in the M.

Definition 4: If s_0 and s_n are situations in M, and $x=<s_0,s_1,...,s_n>$ is a behavior starting at s_0 and ending at s_n in M where n can be $+\infty$, then we inductively define $\models$ as follows:

(S1) $M,s_0 \models P$ if and only if P is true in $L(s_0)$, where P is an atomic proposition.

(S2) $M,s_0 \models$(and p q) if and only if $M,s_0 \models p$ and $M,s_0 \models q$, $M,s_0 \models$(not p) if and only if it is not the case that $M,s_0 \models p$,

(B1) $M,x \models p$ if and only if $M,s_0 \models p$,

(B2) $M,x \models$(and p q) if and only if $M,x \models p$ and $M,x \models q$, $M,x \models$(not p) if and only if it is not the case that $M,x \models p$,

(B3) $M,x \models$(strong-until p q) if and only if there is a nonnegative integer $i \leqq n$, such that $M,x^i \models q$ and for every nonnegative integer $j<i$, $M,x^j \models p$, $M,x \models$(next p) if and only if n=0 or $M,x^1 \models p$.

Some of the definitions of temporal operators based on the semantics are given bellow.

(eventually p)$\equiv$(strong-until true p)
(always p)$\equiv$(not (eventually (not p))) (5)
(strict-precedes p q)$\equiv$
(and (not q) (strong-until (not (next q)) p))
(almost-everywhere p)$\equiv$(eventually (always p))

When the behavior tree M is finitely closed, but contains feedback cycles or quiescences, the length of an $x \in M$ can be infinite, i.e., $n=+\infty$, and consequently the (B3) becomes not to be computable. In this case, the partial unwinded tree $\overline{M}(\phi)$ derived from M must be used for the above semantics in stead of M itself. The strict procedure to derive the $\overline{M}(\phi)$ is detailed in [Kuipers 1994a]. Briefly speaking, a behavior $x \in \overline{M}(\phi)$ has twice unwinded cycles of a behavior loop in maximum together with once unwinded cycles of some other behavior loops. Otherwise, the behavior x has a path to one of quiescences together with once unwinded cycles of some other behavior loops. The $\overline{M}(\phi)$ has the minimum size to check the (B3).

The HOE handles a situation diagram D as same as the other standard envisioning processes [de Kleer and Brown 1984, Kuipers 1994b]. The situation diagram D is a finite graph having finite number of situations and transition arcs. This can be easily rewritten into closed behavior tree [Kuipers 1994b]. For the purpose to use the EBTL in the HOE, we define three types of conversion operations on D.

Definition 5: Given a situation diagram D, an starting situation s_0 and an ending situation s_n,

(C1) An operation $C_1(D,s_n) \rightarrow M_1$ is to enumerate every inversely reachable situations and transition arcs from the s_n in D and form a behavior tree M_1 while truncating the tree at the situations tagged as "cycle" or "quiescent". The inverse reachability means the reachability in the diagram D' where the direction of every transition arc in D is inverted.

(C2) An operation $C_2(D,s_0,s_n) \rightarrow M_2$ is to enumerate every behaviors from the s_0 to the s_n in D and form a behavior tree M_2 while truncating the tree at the situations tagged as "cycle".

(C3) An operation $C_3(D,s_0) \rightarrow M_3$ is to enumerate every reachable situations and transition arcs from the s_0 in D and form a behavior tree M_3 while truncating the tree at the situations tagged as "cycle" or "quiescent".

Figure 2 (a) is the total envisionment of a damped oscillation system. The $C_1(D,s_n)$ derives the ordinary attainable envisionment M_1 to s_n as depicted in fig.2 (b). The $C_2(D,s_0,s_n)$ gives a cycle structure as shown in fig.2 (c). $C_3(D,s_0)$ also derives a cycle with quiescences in fig.2 (d).

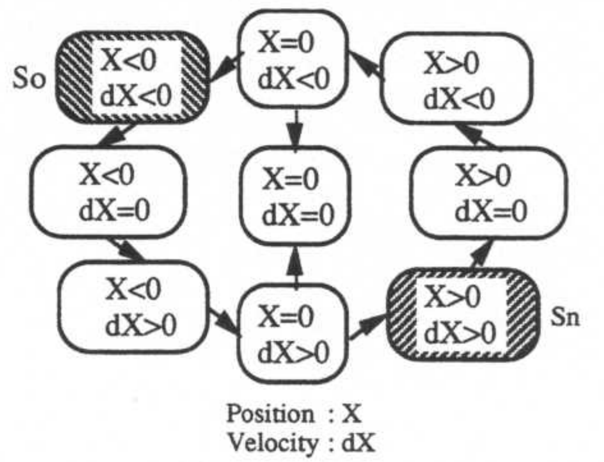
Now the three types of "EBTL-filters" are defined based on the above notions. Let D_x be a union of S_x and R_x, where S_x is a set of situations in a behavior x in the M or $\overline{M}(\phi)$, and R_x is a set of situation transitions within the x.

Definition 6: Given a EBTL-token p, a situation diagram D, $S_0=\{s_{0i} \mid \text{each } s_{0i} \text{ is a starting situation in D}, i=1,\ldots,h.\}$ and $S_n=\{s_{nj} \mid \text{each } s_{nj} \text{ is an ending situation in D}, j=1,\ldots,k.\}$,

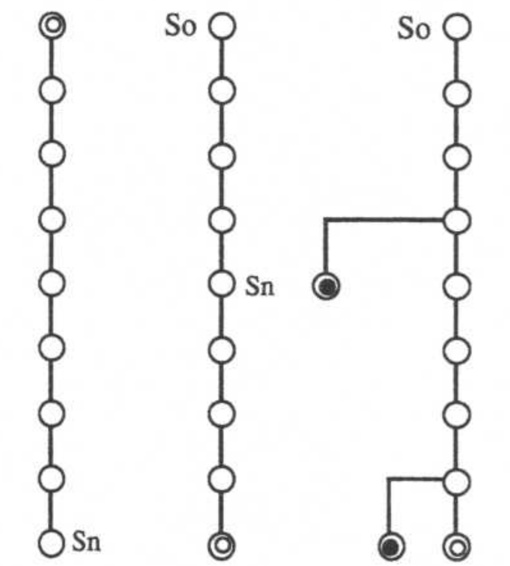
(F1) A filter $F_1(D,S_n,p) \rightarrow D_1$:
When p=(possibly q) where q is a proposition,
if $\exists x \in M_{1j}$, $\exists s_{nj} \in S_n$ that $C_1(D,s_{nj}) \rightarrow M_{1j}$ and $M_{1j},x \models q$,
then D_1=D, else $D_1=\phi$.
When p=(necessarily q) where q is a proposition,
$D_1=\cup D_x$ for all $x \in M_{1j}$ and $s_{nj} \in S_n$ that $C_1(D,s_{nj}) \rightarrow M_{1j}$ and $M_{1j},x \models q$.

(F2) A filter $F_2(D,S_0,S_n,p) \rightarrow D_2$:
When p=(possibly q) where q is a proposition,
if $\exists x \in M_{2ij}$, $\exists s_{0i} \in S_0$ and $\exists s_{nj} \in S_n$ that $C_2(D,s_{0i},s_{nj}) \rightarrow M_{2ij}$ and $M_{2ij},x \models q$,
then D_2=D, else $D_2=\phi$.
When p=(necessarily q) where q is a proposition,
$D_2=\cup D_x$ for all $x \in M_{2ij}$, $s_{0i} \in S_0$ and $s_{nj} \in S_n$ that $C_2(D,s_{0i},s_{nj}) \rightarrow M_{2ij}$ and $M_{2ij},x \models q$.

(F3) A filter $F_3(D,S_0,p) \rightarrow D_3$:
When p=(possibly q) where q is a proposition,
if $\exists x \in M_{3i}$ and $\exists s_{0i} \in S_0$ that $C_3(D,s_{0i}) \rightarrow M_{3i}$ and $M_{3i},x \models q$,
then D_3=D, else $D_3=\phi$.
When p=(necessarily q) where q is a proposition,
$D_3=\cup D_x$ for all $x \in M_{3i}$ and $s_{0i} \in S_0$ that $C_3(D,s_{0i}) \rightarrow M_{3i}$ and $M_{3i},x \models q$.



(a) total envisionment



(b) M_1 (c) M_2 (d) M_3

Figure 2 Conversion from a situation diagram to behavior trees for a damped oscillation system.

3 Faster HOE Under Practical Condition

3.1 Algorithm

The outline of the HOE is depicted in fig.3. The vertical direction from the top to the bottom of the box stands for the time evolution of the behaviors and actions of an objective system. The horizontal axis represents the size of the set involving assumptions of QPT-tokens and constraints of EBTL-filtering. The shadowed area is the input information to the HOE, while the white part is its output. The HOE uses the temporal assumptions Ps_k and the temporal EBTL-tokens Es_k for each time interval or instant of a partial situation. The HOE also uses the set of the permanent assumptions, Pf, specified by the domain model and the fixed part of the scenario throughout the entire envisioning. Pf corresponds to the ordinary scenario's part excluding the partial history. The HOE derives situation nodes allowed within sound combinations of the remaining opened assumptions, and removes any inconsistent nodes with the specifications in Es_k by the EBTL-filters. Accordingly, the HOE focuses on only the situations of the objective system within the intentionally specified partial behaviors and actions.

A new version of the HOE algorithm is represented in fig.4. The basic idea of this algorithm is from the work of the total envisioning done by Forbus [Forbus 1988]. The original version of the HOE derives sound behaviors subject to the given partial history based on the multiple attainable envisionings [Washio 1994]. On the other hand, the new version incrementally perform a ATMS-based total envisioning (or action-augmented envisioning, if actions must be take into account.) for each partial situation, and filters only the situation transitions from the preceding partial situation to the succeeding. The reasoning speed of this algorithm is expected to be faster than the former one due to the high efficiency of ATMS, if the size of the total envisionment for each partial situation is not very large. As the essential structure of this algorithm is independent from the aforementioned EBTL-Filter, the conventional HOE excluding EBTL-tokens can be performed by specifying every Es_k $(k=1,\cdots,n)$ as a vacuous.

(step 1) is to enumerate all possible situations and their transitions for the first partial situation. The total envisioning (or action-augmented envisioning) under the conditions of Pf and Ps_1 is performed. If the first partial situation is not consistent with the Pf, then no solutions are obtained, and the process is halted.

(step 2) is to identify all possible one step transitions from situations in the current partial situation. The one step attainable (or attainable action-augmented) envisioning is an ordinal attainable (or attainable action-augmented) envisioning under a given initial situation, s_{ki}, but its calculation is limited to one situation transition. The notation, Initial(s_{ki}), expresses that s_{ki} is a given initial condition for the envisioning.

(step 3) first enumerates all possible situations and their transitions for the next partial situation, and identifies every

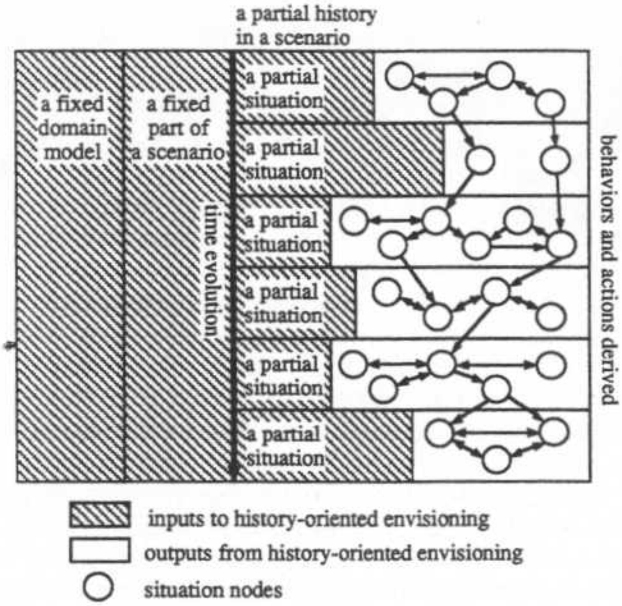


Fig.3 The outline of the history-oriented envisioning.

(step 1) $k \leftarrow 1$. $D \leftarrow \phi$. Perform Subprocess(k).

(step 2) For all $s_{ki} \in S_k$, let the set of assumptions P'_{ki} be $Pf \cup$ Initial(s_{ki}), and perform one step attainable (or attainable action-augmented) envisioning under P'_{ki}, respectively. Let the newly obtained set of situations, S'_k, situation transitions, R'_k, and $D'_k = S'_k \cup R'_k$.

(step 3) Perform Subprocess(k+1). Let S_{nk} be $S_{k+1} \cap S'_k$, R_{nk} be $\{r \mid r(\in R'_k)$ is a transition to a $s \in S_{nk}\}$, and $D_{nk} = S_{nk} \cup R_{nk}$. If $D_{nk} = \phi$ then stop.

(step 4) If k=1 then {
    If $Es_k = \phi$ then
        $Df_k = \cup D_x$ for all $x \in M_{1kj}$ and $s_{nkj} \in S_{nk}$ that $C_1(D_k \cup D_{nk}, s_{nkj}) \rightarrow M_{1kj}$.
    else perform EBTL-filter
        $Df_k = \cap F_1(D_k \cup D_{nk}, S_{nk}, p)$ for all $p \in Es_k$.}
else {
    If $Es_k = \phi$ then
        $Df_k = \cup D_x$ for all $x \in M_{1kij}$, $s_{oki} \in S_{ok}$ and $s_{nkj} \in S_{nk}$ that $C_2(D_k \cup D_{ok} \cup D_{nk}, s_{oki}, s_{nkj}) \rightarrow M_{1kij}$.
    else perform EBTL-filter
        $Df_k = \cap F_2(D_k \cup D_{ok} \cup D_{nk}, S_{ok}, S_{nk}, p)$ for all $p \in Es_k$.}
If $Df_k = \phi$ then stop.

(step 5) $D \leftarrow D \cup Df_k$, $D_{ok+1}(=S_{ok+1} \cup R_{ok+1}) \leftarrow Df_k \cap D_{nk}$, $k=k+1$.
If k<n then go to (step 2),
else {
    If $Es_k = \phi$ then
        $Df_k = \cup D_x$ for all $x \in M_{1ki}$ and $s_{oki} \in S_{ok}$ that $C_3(D_k, s_{oki}) \rightarrow M_{1ki}$.
    else perform EBTL-filter
        $Df_k = \cap F_3(D_k, S_{ok}, p)$ for all $p \in Es_k$.
    If $Df_k = \phi$ then stop else $D \leftarrow D \cup Df_k$, end.}

Subprocess(k) {
    Let the set of assumptions P_k be $Pf \cup Ps_k$.
    Perform a total (or action-augmented) envisioning under P_k, and let D_k be the resultant envisionment $S_k \cup R_k$ where S_k is a set of situations, and R_k is a set of situation transitions. If $D_k = \phi$ then stop.}

Fig. 4 An algorithm of faster HOE.

one step transition from the current partial situation to the next. When any current situations can not transit to the next partial situation, the process is halted.

(step 4) is a filtering process. When any EBTL-tokens are not specified in a current partial situation, the reachability of the situation transitions from the preceding partial situation to the succeeding through the current is tested, and the part of the total envisionment satisfying this reachability is filtered. If some EBTL-tokens are specified, then the EBTL-filters are also used to filter the situation transitions. For the first partial situation (k=1), only the reachability to the succeeding partial situation is checked, because its preceding does not exist.

(step 5) simply accumulates the resultant envisionment for each partial situation into D, and revise some data for the envisioning in the next step. For the final partial situation (k=n), the filtering of the situation transitions which is attainable from the preceding is performed, and final result of the HOE is accumulated in D. If any situations are not filtered in the (step 4) or (step 5), the process is halted.

The most of the computational load in this algorithm is caused by the Subprocess in the (step 1) and (step 3) performing total envisioning for the opened assumptions in each partial situations. The load of this step strongly depends on the number of possible situations for each partial situation. The number of possible situations rapidly decreases almost exponential to the number of QPT-tokens specified in the background scenario and the partial situation. Hence, the computational load will be efficiently reduced, when many specifications are included in each partial situation. As the computational efficiency of the ATMS-based total envisioning is quite higher than the TMS-based attainable envisioning to envision sound behaviors of a system [Forbus 1988], the proposed new algorithm is expected to be advantageous for applications to specify many tokens in every partial situations. The loads of the other steps are not very significant. (step 2) to perform only one step reasoning for each situation transition is a quite cheap process. (step 3) is merely a simple set operation. The filtering process of (step 4) will become heavy in some degree, if the size of the envisionment obtained in (step 1) and/or (step 3) is large. However, the load of this step may be negligible, when many specifications are included in each partial situation. The simplicity of (step 5) is trivial except the final partial situation, and the load is basically same with the (step 4) for the final.

An advantage of this algorithm is that the conventional total envisioning [Forbus 1988, 1989] can be utilized as parts of its process while reducing their solutions and processing time based on the information in a partial history. The unique difference of the envisioning utilized here from the conventional total envisioning is the introduction of some constraints associated with situation generations. If a partial situation is an partial event, then every situation in the partial situation must involve limit hypotheses. On the contrary, they should not involve any limit hypotheses, when they are in an partial episode. The detailed constraints are described in [Washio 1994]. Another advantage is its incremental structure to process a partial history which enables its on-line application to import the new partial situation information step by step. This feature is expected to be profitable for the practical applications of control, planning, measurement interpretation and diagnosis.

3.2 Soundness and Complexity

The standard total (or action-augmented) envisioning derives sound behaviors and actions of a system under closed world assumptions [Forbus 1988]. The standard attainable (or attainable action-augmented) envisioning is also sound subject to its possible initial conditions under the same assumptions. Hence, each standard envisionment generated in (step 1), (step 2) and (step 3) in the algorithm depicted in fig.4 is sound for the given assumptions. The other steps of (step 4), (step 5) and a part of (step 3) reduce the generated situation nodes. Among these two steps, (step 4) and (step 5) are clearly sound, because they just filter situation nodes consistent with the constraints required for the satisfaction of the EBTL-tokens and the transitions from the preceding situation to the succeeding as well as the standard envisioning internally does. (step 3) is also sound, since it maintains all transitions from the current to the next. These observations support the soundness of the HOE conducted through the algorithm of fig.4 under the closed world assumptions.

The complexity of an envisioning process sensitively depends on the number of unspecified assumptions for an envisionment [Forbus 1988, 1989]. Let P be the set of assumptions for a scenario, where its fixed portion is $Pf \subseteq P$. The set of unspecified assumptions for the standard envisioning is Pu=P-Pf. If Pu consists of pairs of independent propositions p and $\neg p$, the upper bound of states number could increase by $O(2^{|Pu|-1})$. On the other hand, each partial situation specifies some extra portion of P in the HOE. The part of unspecified assumptions in P with respect to each partial situation k is $Pu_k = P-(Pf \cup Ps_k)$. Hence, the upper bound of the complexities of the total envisionings in (step 1) and (step 3) of fig.4 for each partial situation are almost $O(2^{|Pu_k|-1})$, respectively, and therefore the upper bound of the entire complexity of this HOE algorithm will be approximately $\sum_{k=1}^{n} O(2^{|Pu_k|-1})$, because the total envisionings in (step 1) and (step 3) are the major source of the complexity. As the number of partial situations in a partial history is independent with the assumptions, and also each $|Pu_k|$ is less than $|Pu|$, the complexity of this HOE can be quite small comparing with the standard total envisioning. On the other hand, this upper bound of the complexity is almost same with that of the original HOE algorithm [Washio 1994].

Usually, the computational load is not proportional to the upper bound of the complexity, because the physical and temporal constraints suppress their complexities. In addition, the limited number of the initial situation transiting from the preceding partial situation reduces the complexity of the envisioning in case of the attainable envisioning based HOE. Also, the use of ATMS as same as the standard total envisioning provides efficient computation in case of the total envisioning based HOE. Accordingly, the advantage on the computational load of these methods will vary depending on the physical and temporal constraints of every system to be envisioned.

4 An Example

The performance of the proposed HOE introducing temporal logic has been tested through the application to the control of a steam generator presented in our preceding work [Washio 1994] for the comparison with the previous HOE version. Figure 5 depicts the overview of the steam generator. It has a primary water tube (p-tube) passing through a secondary boiler tank (s-tank). Highly pressured hot water is supplied from a primary heat source by a pump. When the temperature of the primary water (p-water) is higher than the boiling point of the secondary water (s-water) in the low pressure tank, the heat flow from the primary to the secondary side can boil the secondary water. To compensate the decrease of the secondary water amount due to the escape of the steam (s-steam) to a turbine generator, the extra water feed (f-water) to the tank through a feed pipe (f-pipe) is required. At the beginning of its operation, the boiling of the secondary water has not occurred yet. We could qualitatively determine the future change of the primary water flow rate and its temperature based on the operational conditions of the heat source and the primary pump in the upper stream. Also, the future change of the temperature of the secondary feed water is qualitatively known based on the information of its reservoir. The temperatures of p-water and f-water are supposed to increase monotonically, while the flow rate of p-water are predicted to decrease monotonically in the mean time, and three of them are considered to settle at certain levels after some time. Our task is to plan all sound control strategies of the secondary water feed to the tank to start the boiling, when the three boundary quantities finish their transients. This kind of model based planing tasks have been researched in many AI literatures [Dean and Siegle 1990; Drabble 1993]. But, the most of them utilize the repetition of the attainable envisioning and its evaluations. In contrast with such conventional solution search, the HOE enumerates all possible plans within a finite number of envisionings.

A possible partial history corresponding to our mission is shown in fig.6. It specifies the intended behaviors of the steam generator together with the predicted disturbances exogenously driven. The occurrence of boiling of the secondary water is intended at the final stage of the transients. Figure 7 represents two partial situations in the partial history. The former specifies the initial situation associated with the three boundary quantities, an endogenous quantity, i.e. temperature-of(s-water), and the intended processes. It involves an EBTL-token (represented in italics) stating that the flow rate of feedwater must be in (0,Ffmax) initially and achieve Ffmax finally within this partial situation. The latter specifies that the endogenous temperature-of(s-water) reaches at its boiling point, and simultaneously the boiling process is activated, when the three boundary quantities reach to their goal levels while maintaining the amount-of(s-water), the heat-flow and the fluid-flow. The two EBTL-tokens says that the amount of secondary water must not undershoot, before it becomes stable, and the flow rate of feedwater must be within (0,Ffmax), until the amount of secondary water becomes stable. If any EBTL-tokens were not used to specify the identical behaviors, more granular partial situations must be used.

Figure 8 depicts the result of the envisioning under the partial history of fig.6 excluding all EBTL-tokens. Totally, 29 situations were found, and this is identical result with the former work [Washio 1994]. Both of the new and original version of the program were tested on C++ with the SPRAC-10 and Solaris Operating System. The total computation time was 58sec for the proposed new version, and in contrast, it was 615sec for the original. The new

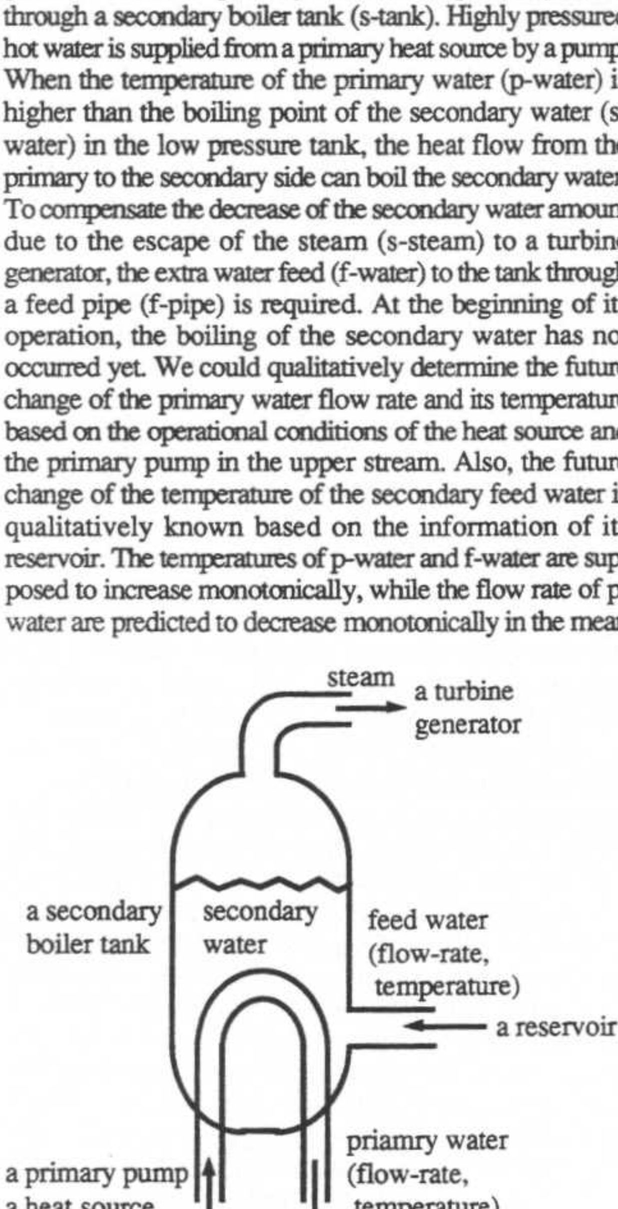


Fig. 5 A steam generator in a power plant for electricity generation.

Partial History Boiling-Control
Partial Slices: (T Initial-State(I0) I0)
(T Start-of-Transient(I1) I1)
(T Monotonic-Transient(I2) I2)
(T End-of-Transient-and-Start-of-Boiling(I3) I3)
(T Final-State(I4) I4)
Time Constraints: (start(I0)=end(I0)), (end(I0)=start(I1)),
(start(I1)=end(I1)), (end(I1)=start(I2)),
(start(I2)<end(I2)), (end(I2)=start(I3)),
(start(I3)=end(I3)), (end(I3)=start(I4)),
(start(I4)=end(I4))

Fig. 6 A partial history to control the boiling of secondary water.

Partial Situation Initial-State(?time)
Individuals: p-tube a pipe
f-pipe a pipe
s-tank a container
p-water a contained liquid
s-water a contained liquid
f-water a contained liquid
Quantities: (T A[temperature-of(p-water)]
>A[temperature-of(f-water)] ?time)
(M A[temperature-of(p-water)] ?time) = Tpmin
(M Ds[temperature-of(p-water)] ?time) = 0
(M A[temperature-of(f-water)] ?time) = Tfmin
(M Ds[temperature-of(f-water)] ?time) = 0
(T A[temperature-of(s-water)]
<A[t-boil(s-water)] ?time)
(M Ds[temperature-of(s-water)] ?time) = 0
(M A[flow-rate-of(p-water)] ?time) = Fpmax
(M Ds[flow-rate-of(p-water)] ?time) = 0
(necessarily
(strong-until A[flow-rate-of (f-water)]=(0,Ffmax)
(always A[flow-rate-of(f-water)]=Ffmax)) ?time)
Views:
Processes: (T Status(Heat-flow(p-water, s-water,
p-tube), Active) ?time)
(T Status(Fluid-flow(f-water, s-water,
f-pipe), Active) ?time)
(T Status(Boiling(s-water, Heat-flow),
Inactive) ?time)
Actions:

Partial Situation End-of-Transient-and-Start-of-Boiling(?time)
Individuals: p-tube a pipe
f-pipe a pipe
s-tank a container
p-water a contained liquid
s-water a contained liquid
f-water a contained liquid
Quantities: (T A[temperature-of(p-water)]
>A[temperature-of(f-water)] ?time)
(M A[temperature-of(p-water)] ?time)= Tpmax
(M Ds[temperature-of(p-water)] ?time) = 1
(M A[temperature-of(f-water)] ?time)= Tfmax
(M Ds[temperature-of(f-water)] ?time) = 1
(T A[temperature-of(s-water)]
=A[t-boil(s-water)] ?time)
(M Ds[temperature-of(s-water)] ?time) = 0
(M A[flow-rate-of(p-water)] ?time) = Fpmin
(M Ds[flow-rate-of(p-water)] ?time) = -1
(M A[amount-of(s-water)] ?time) = (0,Msmax)
(necessarily
*(strict-precedes Ds[*amount-of(s-water)*=-1*
*(always Ds[*amount-of(s-water)*=0)) ?time)*
(necessarily
(strong-until A[flow-rate-of (f-water)]=(0,Ffmax)
*(always Ds[*amount-of(s-water)*=0)) ?time)*
Views:
Processes: (T Status(Heat-flow(p-water, s-water,
p-tube), Active) ?time)
(T Status(Fluid-flow(f-water, s-water,
f-pipe), Active) ?time)
(T Status(Boiling(s-water, Heat-flow),
Activated) ?time)
Actions:

Fig. 7 An example of a partial slice
for the control of the secondary water boiling.

algorithm works around 10 times faster in this example. This may be due to the specifications of many tokens in each partial situations. As the sizes of the total envisionments are limited under this condition, the relatively efficient ATMS-based algorithm derives higher speed performance comparing with the original one. Figure 9 shows comparisons of the computation time between the new and the original algorithms at various numbers of tokens specified in partial histories. The tokens were randomly chosen to be specified. The speed of the new algorithm is quite higher than the original, when many tokens are specified. On the contrary, it rapidly becomes slow relative to the original one, when fewer tokens are specified. This feature of the new algorithm is considered to be advantageous for the control and diagnosis of process plants, because the most part of the behaviors and actions can be exogenously specified by our operational goal and sensors information in those applications.

Figure 10 shows the results for two parts of the history oriented envisionment obtained by the partial history involving the EBTL-tokens. Both results of "Initial-State" and "End-of-Transient-and-Start-of-Boiling" follow the contextual specifications by the EBTL-tokens, and more specific and realistic control strategies have been obtained.

5 Discussions and Related Works

The work presented in this paper extended the knowledge representation of partial history from the primitive and snapshot facts to the contextual facts by introducing a temporal logic. As the temporal logic is highly expressive, it can specify qualitative but contextually very complicated behaviors. Drabble developed a system named as EXCALIBUR for planning and reasoning with process systems [Drabble 1993]. The system utilizes some attainable envisioning processes, and manages the actions to change continuous process quantities not only the ones to cause discontinuous change of views and processes. But, it does not handle the explicit specifications on the behaviors evolved in process systems in the envisioning. Though it can take a tree and hierarchical structure of actions sequences, they are limited to finite behaviors, and each step of the behaviors must be specified explicitly. In contrast, the framework of temporal logic in our work enables the comprehensive specifications of infinite, implicit and abstracted behaviors and actions. This extends the uniqueness of qualitative envisioning in practical applications, since almost no other approach can handle such abstracted specifications. The original idea to introduce temporal logics to qualitative reasoning was proposed by Kuipers [Kuipers 1994]. He presented an idea to use the EBTL to validate if a theorem written in the logic holds for any ordinary differential equation consistent with the qualitative differential equation that generated the QSIM behavior tree. On the contrary, our work utilize the EBTL for the behavior generation consistent with the logic under a given scenario in the framework of QPT.

Another major characteristic is the better efficiency

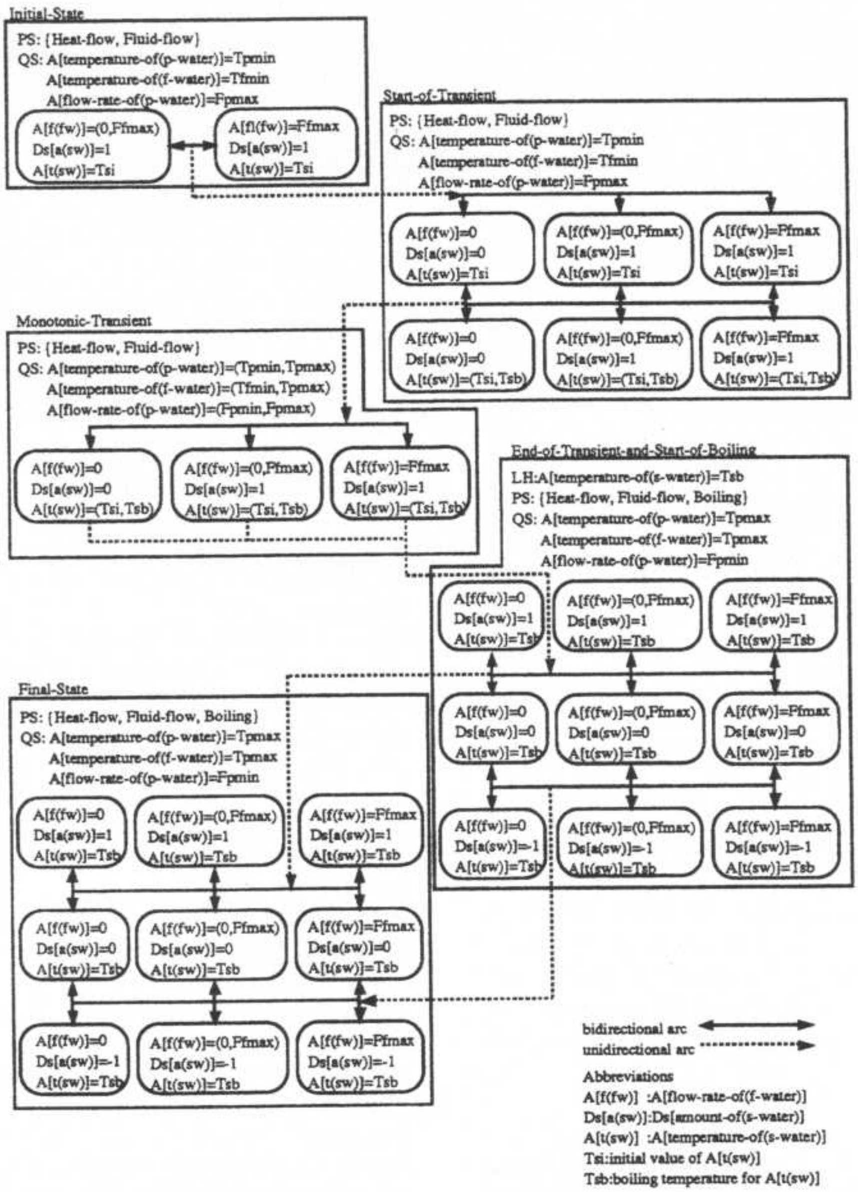
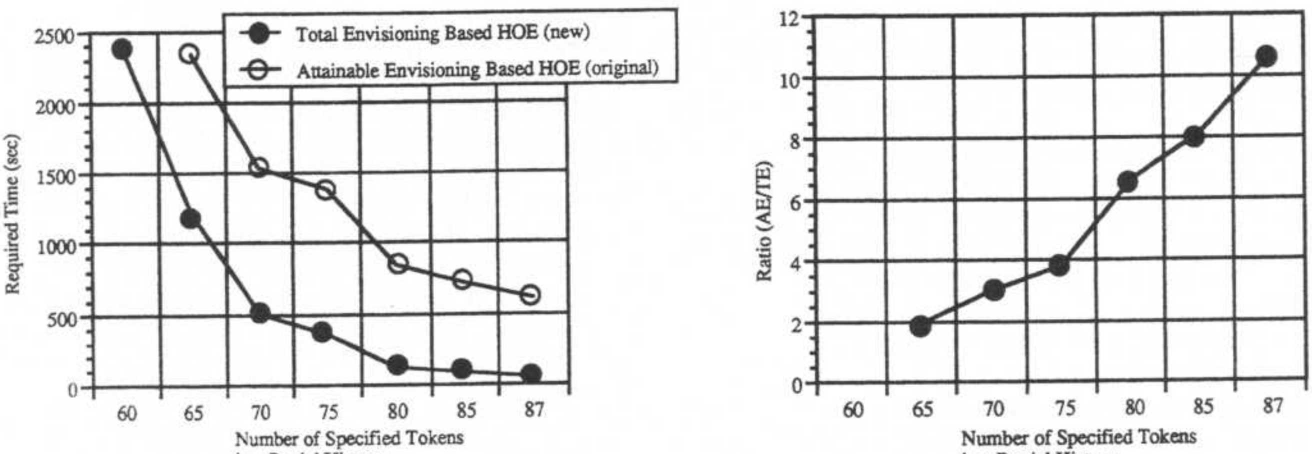
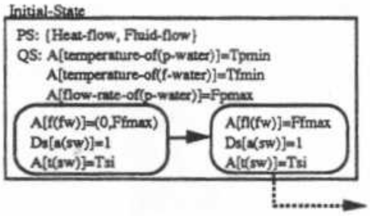


Fig.8 A situation transition diagram of a steam generator for a partial history without EBTL-tokens.

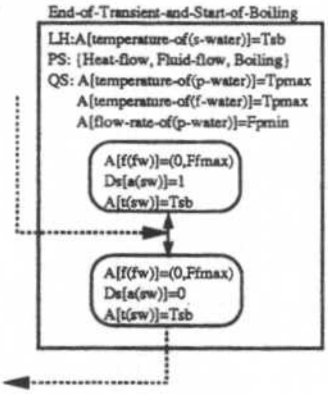


(a) Comparison of computation time (b) Ratio of computation time

Fig.9 Comparison of computation time between total envisioning based and attainable envisioning based algorithms.



(a) A partial situation:Initial-State



(b) A partial situation: End-of-Transient-and-Start-of-Boiling

Fig.10 Parts of a situation transition diagram of a steam generator for a partial history involving EBTL-tokens.

of envisioning under the specification of many tokens in a partial history. This has done by using ATMS-based total envisioning dominantly instead of TMS-based attainable envisioning. The basic idea is from the work by Forbus [Forbus 1988]. He compared the computation speed of both approaches, and obtained the result that the former is about 13 times faster in the average. Our result is almost consistent with his evaluation.

6 Conclusion

A new fast algorithm of the history-oriented envisioning (HOE) enabling the introduction of the temporal logic has been proposed in this work. The potential applicability and efficiency of this method to some realistic conditions have been readily confirmed through an example of a control strategy planning. The major characteristic of the new HOE are summarized as follows.

(1) Flexible specification of intentional partial behaviors and actions in the HOE by introducing a temporal logic.
(2) Small complexity and good efficiency comparing with the conventional envisioning and the former HOE under the exogenous specification of many tokens.
(3) Incremental envisioning structure to import the assumptions in an on-line manner.

The ideas presented here will extend the qualitative envisioning theory toward its application to practical tasks of simulation, planning, design, measurements interpretation, control and diagnosis.

References

[Amador et al. 1993] Franz G. Amador, Adam Finkelstein and Daniel S. Weld. Real-Time Self-Explanatory Simulation. In proceedings AAAI-93, pp.562-567, 1993.

[Caloud 1987] Philipe Caloud. Toward Continuous Process Supervision. In proceedings IJICAI-87, pp. 1086-1089, 1987.

[Dean and McDermott 1987] Thomas Dean and Drew V. McDermott. Temporal Database Management. Artificial Intelligence, 36(3):375-399, 1988.

[Dean and Siegle 1990] Thomas Dean and Greg Siegle. An Approach to Reasoning About Continuous Change for Applications in Planning. In proceedings AAAI-90, pp.132-137, 1990.

[de Kleer and Brown 1984] Johan de Kleer and John Seely Brown. A Qualitative Physics Based on Confluence. Aritificial Intelligence, 24:7-83, 1984.

[Drabble 1993] Brain Drabble. EXCALIBUR: A Program for Planning and Reasoning with Process. Aritificial Intelligence, 62:1-40, 1993.

[Forbus 1984] Kenneth D. Forbus. Qualitative Process Theory. Aritificial Intelligence, 24:85-168, 1984.

[Forbus 1988] Kenneth D. Forbus. QPE: Using Assumption-based Truth Maintenance for Qualitative Simulation. Artificial Intelligence in Engineering, 3(4):200-215, 1988.

[Forbus 1989] Kenneth D. Forbus. Introducing Actions into Qualitative Simulation. In proceedings IJCAI-87, pp.1273-1278, 1989.

[Forbus and Falkenhainer 1990] Kenneth D. Forbus and Brain Falkenhainer. Self-explanatory Simulations: An Integration of Qualitative and Quantitative Knowledge. In proceedings AAAI-90, pp. 380-387 1990.

[Forbus and Falkenhainer 1992] Kenneth D. Forbus and Brain Falkenhainer. Self-explanatory Simulations: Scaling Up to Large Models. In proceedings AAAI-92, pages 685-690, 1992.

[Kuipers 1984] Benjamin Kuipers. Commonsense Reasoning about Causality: Deriving Behavior from Structure. Aritificial Intelligence, 24:169-203, 1984.

[Kuipers 1986] Benjamin Kuipers. Qualitative Simulation. Aritificial Intelligence, 29:289-338, 1986.

[Kuipers 1994a] Benjamin J. Kuipers and Benhamin Shults. Reasoning in Logic about Continuous Systems. A working paper of the Eighth International Workshop on Qualitative Reasoning about Physical Systems: QR'94, 164-175, 1994.

[Kuipers 1994b] Benjamin J. Kuipers. *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge*. Cambridge, MA: MIT Press, 1994.

[Hayes 1979] P. J. Hayes. The naive physics manifesto, in: D. Michie (Ed.), Expert Systems in the Electric Age (Edinburgh University Press, Edinburgh, 1979).

[Washio 1994] Takashi Washio. A History-oriented Envisioning Method. A working paper of the Eighth International Workshop on Qualitative Reasoning about Physical Systems: QR'94, 286-294, 1994.

[Williams 1984] Brian C. Williams. Qualitative Analysis of MOS Circuits. Aritificial Intelligence, 24:281-346, 1984.