A History-oriented Envisioning Method

Takashi Washio, Ph.D.
Research Center for Safety Engineering
Mitsubishi Research Institute, Inc.
1-8-1, Shimomeguro, Meguro-ku,
Tokyo 153, Japan
washio@mri.co.jp

Abstract

A novel and generic approach named as "history-oriented envisioning" is proposed to qualitatively envision all possible and sound situations focusing on our intended partial behaviors and actions of an objective system. Some basic notations called "partial slices" and a "partial history" are also introduced as the extensions of the conventional history and its slices representing qualitative, temporal behaviors and actions of the system. They provide basic information of the intended partial behaviors and actions to the history-oriented envisioning. A major characteristic of the envisioning proposed here is the low complexity reduced by the specified history of behaviors and actions in addition to the conventional scenario. Another important characteristic is its incremental structure enabling the import of the measured information of the objective system in an online mode. These features of the history-oriented envisioning will promote new progress of qualitative envisioning theory toward its application to practical tasks of simulation, planning, design, measurements interpretation and diagnosis. The efficiency of the proposed method is demonstrated through an example to control a steam generator of a large scale plant.

1 Introduction

One of the primary tasks of qualitative reasoning is the envisioning of system "behaviors" established through leading works [de Kleer and Brown 1984; Forbus 1984, 1988; Kuipers 1984, 1986]. The "behaviors" are the collection of possible situation transitions resulted by system operations. The conventional framework of the envisioning consists of "attainable envisioning" and "total envisioning". The former derives the qualitatively possible situation transitions achieved from a specified initial state of a system. The later derives all possible situation transitions that may occur in the operations. The basic idea of these methods is to evaluate possible and sound behaviors of a system while maintaining a set of initially given background assumptions associated with exogenous quantity states, views and processes to the system without imposing our intentional changes of the assumptions.

In contrast with this standard methodology, the author is interested in the envisioning in case that some portions of the evolutional system behaviors are specified in advance by our intention. Some works to introduce exogenously specified quantity, view and process transitions into the envisioning have been done [Forbus 1989; Drabble 1993; Iwasaki et al. 1993; Vescovi et al. 1993]. Forbus defined an "action" as an exogenous replacement of some background assumptions in a system scenario, and established "action-augmented envisioning" that enumerates all possible transitions among combinations of quantity states, views, processes and actions. Besides, Drabble extended the notion of the actions to involve the exogenous specification of quantity states and to have qualitative time intervals. He also introduced a hierarchical sequence of actions to represent complex influences exogenously driven. Iwasaki and Vescovi proposed a language to specify intended functional behaviors in terms of causal transition rules. The latter two studies basically utilize the repetitions of the attainable envisioning to search their intended system behaviors.

However, a difficulty of combinatorial explosion of derived situations in the aforementioned envisioning methods have been reported, especially for the total and action-augmented envisioning, when the methods are adopted to practical scale applications [Caloud 1987; Forbus 1989; Forbus and Falkenhainer 1990, 1992; Amador et al. 1993]. For example, a self-explanatory simulation system "SimGen Mk2" to envision only local states of a system requires 4 hours to compile a simulator for a model of 9 containers and 12 pipes [Forbus and Falkenhainer 1992]. The main cause of this difficulty arises from the vast number of possible states envisioned, e.g. almost 1012 states, even for such a simple system. An efficient remedy to this difficulty is to restrict the scope of the envisioning within the partial behaviors and actions intentionally specified by following our interests or the objectives of application tasks. 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 [DeCoste 1990, 1993; Drabble 1993; Forbus 1986; Forbus and Falkenhainer 1990, 1992; Ishida and Eshelman 1988; Iwasaki 1993; Pearce 1988; Umeda et al. 1991; Yannou 1993]. Their efficiency may be enhanced significantly by introducing the envisioning focused on specific and meaningful behaviors in addition to specific actions.

The work presented here is to propose a novel and generic envisioning method focused on specific partial behaviors and actions of a system so called as "history-oriented envisioning". The efficiency of the proposed method is demonstrated through an example to control a steam generator of a large scale plant in the latter half of this paper.

2 Partial Slice and Partial History

An efficient representation of the partial behaviors and actions which are to be intentionally specified is defined first before the detailed discussion on the history-oriented envisioning. 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" consisting of "episodes" and "events". Events always last for an instant, while episodes usually occur over a time interval. Each episode has a start and an end which are events that serve as its boundaries. Both of an event and an episode can involve the descriptions of quantity states, views, processes, actions, their relations and their transitions at a time (or a time interval). An assertion representing one of such descriptions is called a "token" [Dean and McDermott 1987]. Each token states a primitive fact in an event or an episode such as "amount of water in a pot is 1kg." or "boiling of water occurred.". In addition to these definitions, they also defined a "slice" of a history denoting a piece of a history at a particular time. Thus, a slice can be either of an episode and an event.

Based on these definitions, some new and important ideas on the history are introduced in this work as follows.

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 slice" of a history is either of a partial event and a partial episode.

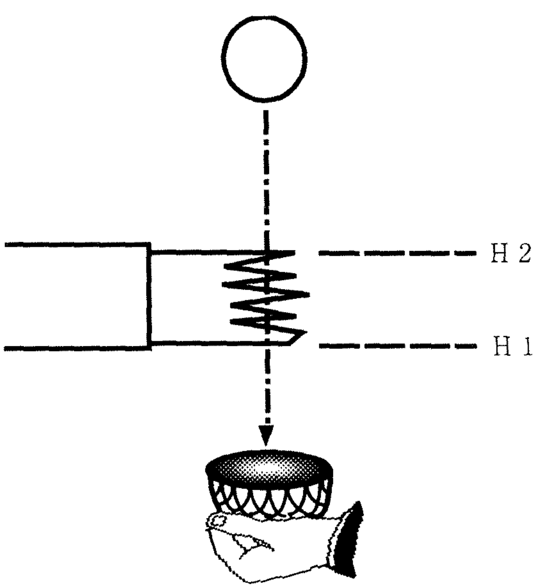

Fig.1 Catching a ball dropped through a flame.

The notation of a partial slice consists of a list of individuals that must exist, a list of quantity values and relations indicating the objects' states, a list of (in)active and (in)activated views, a list of (in)active and (in)activated processes and a list of (in)active and (in)activated actions. The (in)activated views, processes and actions mean these are (in)activated in a partial event, while (in)active ones are to be kept (in)active in a partial slice. The detailed descriptions of the contents in each list follow the notations in QP-theory [Forbus 1984]. T-operators are used to say that a particular token is true at some time, and M-operators represent the measured value of a quantity at some time.

Two of partial slices for the example of catching a ball dropped through a flame depicted in fig.1 are represented as follows.

```
Partial Slice  Heat-Flow-to-Ball-Active(?time)
  Individuals: ball a ball
               flame a flame
  Quantities:  (T A[temperature-of(ball)]
                 <A[temperature-of(flame)] ?time)
               (M A[temperature-of(ball)] ?time) = Tl    (1)
               (M Ds[temperature-of(ball)] ?time) = 1
  Views:
  Processes:   (T Status(Heat-Flow(flame, ball, flame-ball),
                 Active) ?time)
  Actions:

Partial Slice  Catching-Ball-under-Flame(?time)
  Individuals: ball a ball
               flame a flame
               basket a basket
  Quantities:  (M A[position-of(ball)] ?time) = (-∞,H1)
               (M Ds[position-of(ball)] ?time) = -1      (2)
  Views:       (T Status(Contained-Stuff(ball, basket),
                 Activated) ?time)
  Processes:
  Actions:     (T Status(Catch-In(ball, basket),
                 Activated) ?time)
```

The former partial slice represents that the objects of a ball and a flame exist at a particular time. Also, the amount and the derivative's sign of the temperature of the ball are Tl and positive respectively in addition to the fact that it is colder than the temperature of the flame, and the process of heat flow is operating at the same time. The latter means that the objects of a ball, a flame and a basket exist, and the ball descending under the flame is caught and settled in the basket at some time. Some lists and their contents can be left unspecified in a partial slice. For instance, any specifications of the position of the ball are not given within the former partial slice, while the quantity must be determined to specify a unique state of the ball. A distinct partial slice is "No-Specification(?time)" in which all lists are blank. This partial slice is used to represent that any behaviors and actions are unspecified at a time.

The term "?time" represents the temporal specification of a partial slice, and follows the conditions indicated below with respect to its duration and the limit hypotheses involved in its partial slice.

?time is an instant. ⇔ start(?time)=end(?time),
?time is an interval. ⇔ start(?time)<end(?time), (3)
The duration of ?time is unspecified.
⇔ start(?time)≦end(?time).

A partial slice involves some limit hypotheses. (4)
⇒ ?time is an instant.

The condition (4) states that any partial slice involving some limit hypotheses is a partial event. However, a partial event does not have to involve any limit hypothesis, because the extra portion of the event may own some limit hypotheses. On the other hand, any partial episode does not involve any limit hypothesis by definition.

The contents in the lists of individuals and quantities of a partial slice are used as the assumptions for the history-oriented envisioning. For instance, the assertion of "ball a ball" in the individual lists is an assumption specified by the partial slices in the above examples. On the other hand, the views, the processes and the actions in their lists do not represent their assumptions directly. The views and the processes in the scenario and the domain model of the QP-theory have the information of "Individuals", "Preconditions" and "Quantity Conditions" [Forbus 1984]. Also, the actions have the part of "Individuals" [Forbus 1989]. These are their assumptions to be active. Accordingly, the unifications of the contents in the view, the process and the action lists to the scenario and the domain model in the envisioning system provide their assumptions for the partial slice. For instance, the unification of "Heat-Flow(flame, ball, flame-ball)" in the process lists of the partial slice (1) to the following domain model (5) of the heat-flow process serves the contents in the "Individuals", "Preconditions" and "Quantity Conditions" of the process as the corresponding assumptions.

```
Process Heat-Flow(?src, ?dst, ?path)
  Individuals:         ?src an object,
                       Has-Quantity(?src, heat)
                       ?dst an object,
                       Has-Quantity(?dst, heat)
                       ?path a Heat-Path,
                       Heat-Connection(?path, ?src, ?dst)
  Preconditions:       Heat-Aligned(?path)                (5)
  Quantity Conditions: A[temperature(?src)]
                          >A[temperature(?dst)]
  Relations:           Quantity(flow-rate)
                       flow-rate = (temperature(?src)
                                   -temperarture(?dst))
  Influences:          I-(heat(?dst), A[flow-rate])
                       I+(heat(?src), A[flow-rate])
```

Another important assumption in a partial slice is the duration of ?time. Its specification controls the generation of the limit hypotheses in the envisioning process as explained later. The history-oriented envisioning utilizes all of the assumptions described here for a scenario involving the partial slice.

The definition of a "partial history" is given based on the partial slices.

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

A partial history has a list of the T-operators to say that a particular partial slice is true at some time. It also involves a list of constraints following the condition (3) on the temporal relations among the partial slice's time intervals. An example of a partial history for the ball is shown here.

```
Partial History  Initial-and-Final-Ball
  Partial Slices:  (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 No-Specification(I3) I3)
                   (T Position-Decreasing-of-
                     Ball-under-Flame(I4) I4)
                   (T Catching-Ball-under-Flame(I5) I5) (6)
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))
```

The former half of this partial history before No-Specification(I3) specifies the partial behaviors of a ball beginning with its initial position until it touches the flame, while the latter after No-Specification(I3) describes the partial behaviors and actions associated with the ball under the flame until it is caught in a basket. Hence, this partial history specifies two clusters of partial slices mutually apart in time domain. Each partial slice of which the starting and the ending times are identical corresponds to an instant, e.g., Initial-Position-of-Ball(I0) and (T Catching-Ball-under-Flame(I5) I5). The partial slice, (T Heat-Flow-to-Ball-Active(I2) I2), is regarded as any of a partial event and a partial episode, because its duration is not specified in this example. Identical partial slices should not be neighbored mutually, because the neighboring identical partial slices are equivalent to such one partial slice, and they can be merged.

A partial history given to the history-oriented envisioning belongs to a scenario for the envisioning. Some partial histories may not be involved in the possible histories of the scenario. For example, the initiation of the heat flow from the flame to the ball before the touch of the ball to the flame is impossible. When such a partial history is specified, the history-oriented envisioning is halted at its intermediate step, and does not generate any envisionment consistent with the partial history.

3 History-oriented Envisioning

This section describes the outline and the algorithm of the history-oriented envisioning, and discusses its important features.

3.1 Overview of History-oriented Envisioning

The outline of the history-oriented envisioning is depicted in fig.2. 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 spectrum of the assumptions in the envisioning process. The shadowed area is the input information to the history-oriented envisioning, while the white part is its output. The domain

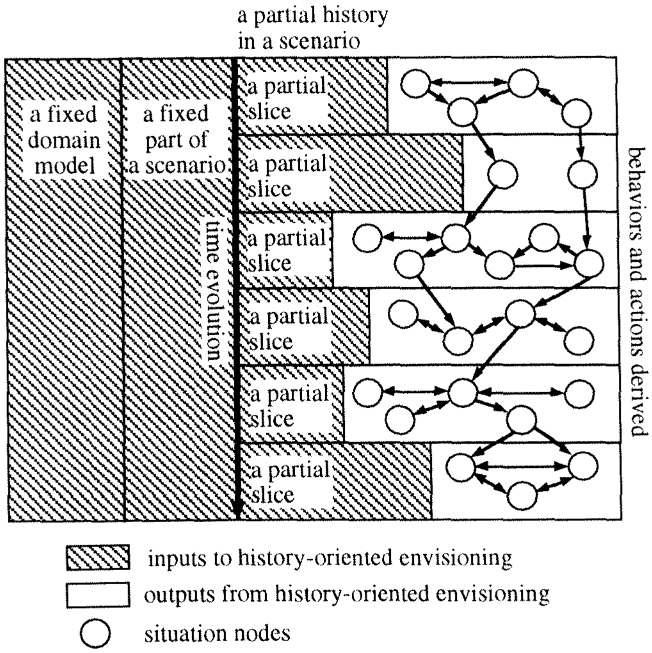

Fig.2 The outline of the history-oriented envisioning.

model and a part of the scenario for the objective system are fixed over the entire time evolution in the envisioning. The conventional envisioning enumerates situation nodes grounded all possible and sound combinations of the rest opened assumptions associated with the system. On the other hand, the history-oriented envisioning imports the specifications on some extra portion of the assumptions for each time interval or instant in form of the partial slice. It derives situation nodes allowed within all possible and sound combinations of the remaining part of the opened assumptions while following the order of the specified time interval or instant. Accordingly, the history-oriented envisioning focuses on only the situations of the objective system within the intentionally specified partial behaviors and actions. The number of the derived situation nodes and the computation amount for the derivation are significantly reduced due to the low ambiguity of the conditions exponential to the number of the opened assumptions.

3.2 Algorithm

A partial history for the history-oriented envisioning must be compiled in advance to derive all assumptions associated with its partial slices. The algorithm of compiling a partial history is depicted in fig.3. This derives the set of assumptions, Psi (i=1,---,n), for every partial slice, PS(i), and its time constraints, TC(i), in the partial history by unifying their views, processes and actions to the scenario and the domain model in the envisioning system. The series of the partial slices and their time constraints in a partial history is sequentially compiled. (step 3) and (step 4) collect the assumptions of a partial slice. (step 3) obtains the assumptions explicitly represented in the individuals lists, the quantities lists and the time constraints of the partial slice, while (step 4) derives implicit ones of the views, processes and actions not directly represented in the partial slice. (step 2) and (step 5) check the violation of the condition (4) in the given partial slices, and quit the compiling if any violations are detected. (step 2)

(step 1) i←1.
(step 2) Choose the ith partial slice PS(i) and the time constraints TC(i) on PS(i).
If any explicit limit hypotheses appear in any lists of the individuals, the quantities, the views, the processes and the actions in PS(i), check the condition (4). If it is violated, then stop.
(step 3) Let Psi be a set of the contents in TC(i), the individuals lists and the quantities lists of PS(i).
(step 4) Unifiy the views, the processes and the actions in their lists to the scenario and the domain model in the envisioning system, and let Psu be a set of the contents in the individuals, the preconditions and the quantity conditions of the unified predicates.
Psi←Psi ∪ Psu.
(step 5) If any implicit limit hypotheses appear in Psi, check the condition (4). If it is violated, then stop.
If i<n, then go to (step 2), else end.

Fig.3 An algorithm of compiling a partial history.

(step 1) i←1. Let the set of assumptions Pi be Pf ∪ Psi.
Perform a total (or action-augmented) envisioning under Pi, and let Q be a set {q(i,j) | q(i,j) is jth situation node generated in the envisioning, and j=1,---,mi.}.
If Q is null, then stop.
(step 2) R←{}.
For j=1 to mi {
(step 2.1) Let the set of assumptions P'i,j be Pf ∪ Initial(q(i,j)).
Perform one step attainable (or attainable action-augmented) envisioning under P'i,j.
(step 2.2) Filter only the situation nodes which satisfies all conditions specified in Psi+1.
Let Rf be a set of the filtered situation nodes, and R←R ∪ Rf. }
If R is null, then stop.
(step 3) Remove every situation node from Q which is not reachable to any r(j) ∈ R (j=1,---,mr).
(step 4) i←i+1. S←{}.
For j=1 to mr {
Let the set of assumptions Pi,j be Pf ∪ Psi ∪ Initial(r(j)).
Perform an attainable (or attainable action-augmented) envisioning under Pi,j.
Let Sf be a set of the situation nodes generated in the envisioning, and S←S ∪ Sf. }
(step 5) S is represented as {q(i,j) | j=1,---,mi.}. Q←Q ∪ S.
If i<n, then go to (step 2), else end.

Fig. 4 An algorithm of history-oriented envisioning.

checks any explicit limit hypotheses appearing in a partial episode, e.g., (T Status(Catch-In(ball, basket), Activated) Interval). On the other hand, (step 5) checks any implicit limit hypotheses appearing in the assumptions of a partial episode unified in (step 4), e.g., (Am[temperature(ball)]=100)&(Ds[temperature(ball)]=1) in the quantitiy conditions of a process.

Once, all assumptions are obtained, they are applied to the algorithm of the history-oriented envisioning represented in fig.4. This algorithm has an incremental structure. It accepts the assumption sets of partial slices in a partial history one by one following their total order described in the time constraints of the history. Pf is the time-independently fixed portion of the set of background assumptions for a scenario over the envisioning. Pf corresponds to the ordinary scenario's part excluding the partial history.

(step 1) is to enumerate all possible situations for the first partial slice. The total envisioning (or action-augmented envisioning, if possible actions must be take into account.) under the conditions of Pf and Ps1 is required, because any preceding situations have not been specified at this step. If the first partial slice is not consistent with the Pf, then no solutions are obtained, and the process is halted.

(step 2) is to identify all situations for the next partial slice which can be directly caused from the current situations. In (step 2.1), all possible one step transitions from a current situation are figured out. The one step attainable (or attainable action-augmented) envisioning is an ordinal attainable (or attainable action-augmented) envisioning under a given initial condition, q(i,j), but its calculation is limited to one situation transition. The notation, Initial(q(i,j)), expresses that q(i,j) is a given initial condition for the envisioning. (step 2.2) filters out any inconsistent situations with the next partial slice. When the next partial slice is the No-Specification(?time), all situations obtained in (step 2.1) is filtered. On the contrary, when any current situations can not transit to consistent situations with the specifications of the next partial slice, the process is halted.

(step 3) is a sort of retrospective reasoning, while the other steps are perspective. Q contains all possible histories from the first to the current partial slice. Because the situations in the current partial slice which can transit to the next are limited as the result of the filtering in (step 2), some preceding histories in Q may not be causatively connected to the next partial slice. This step eliminates such dead end histories in Q.

(step 4) enumerates all possible situations in the new partial slice proceeded from the previous partial slice based on the attainable (or attainable action-augmented) envisioning. Each envisioning starts from a situation figured out in (step 2), and a list of all possible situations evolved for this partial slice is obtained.

(step 5) simply accumulates the situations for the new partial slice in Q. The entire envisioning is ended, and an envisionment following the given partial history is rested in Q, when all partial slices in the partial history have been processed.

The most of the computational load in this algorithm is caused by (step 4). The load of this step strongly depends on the number of its initial states R generated in (step 2), and the number is almost dominated by the efficiency of the situation filtering in (step 2.2) associated with the preceded partial slices. Hence, the computational load will be efficiently reduced, when many specifications are included in each partial slices. The loads of the other steps are not very significant. The total (or action-augmented) envisioning in (step 1) which algorithm is essentially efficient is performed just once, and its processing speed is greatly accelerated by the specifications of the first partial slice, unless the specifications are limited. (step 2) to perform only one step reasoning for each situation transition is also a quite cheap process. (step 3) is merely a network search of which various efficient algorithms are available. The simplicity of (step 5) is trivial.

An advantage of this algorithm is that the conventional total and attainable envisioning [Forbus 1984, 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 one is the imposition of the following rules to the situation node generation associated with the assumption of TC(i). They reject the situation nodes involving any limit hypotheses in a finite time interval.

(start(Ii)=end(Ii))&(The assumptions of
PS(i) do not involve any limit hypotheses.)
$\Rightarrow$ (The assumptions generated for the situation node
in the envisioning must involve some limit hypotheses.) (7)
(start(Ii)<end(Ii)),
$\Rightarrow$ (The assumptions generated for the situation node
in the envisioning must not invole any limit hypotheses.),
where Ii is an instant or an interval for PS(i).

Another advantage is its incremental structure to process a partial history which enables its on-line application to import the new partial slice information step by step. Especially, when the amount of the specifications in each partial slice is large under its on-line import, its computation time will be applicable to the real time processing. These features of the algorithm are expected to be highly profitable for the real time applications of control, planning, measurement interpretation and diagnosis.

3.3 Soundness and Complexity

The standard total (or action-augmented) envisioning is sound for all possible system behaviors and actions under closed world assumptions of which the members are the only possible assumptions for the scenario [Forbus 1988]. The standard attainable (or attainable action-augmented) envisioning is also sound for its possible initial conditions under the closed world assumptions. Hence, each standard envisionment generated in (step 1), (step 2.1) and (step 4) in the algorithm depicted in fig.4 is sound for the given assumptions. The other steps of (step 2.2) and (step 3) reduce the generated nodes. Among these two steps, (step 2.2) is clearly sound, because it just filters situation nodes consistent with the constraints required for the transitions from the current slice to the next as well as the standard envisioning internally does. (step 3) is also sound, since it keeps all histories which do not contradict the assumptions of any partial slices and the scenario's fixed part in the context of a given partial history. These observations support the soundness of the history-oriented envisioning 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, because it does not utilize any information of a partial history. If Pu consists of pairs of independent propositions p and $\neg$p, the number of states could increase by $O(2^{|Pu|-1})$. On the contrary, each partial slice specifies some extra portion of P in the history-oriented envisioning. The part of unspecified assumptions in P with respect to the first partial slice is $Pu1=P-Pf \cup Ps1$. Hence, the complexity of the total envisioning in (step 1) of fig.4 is propotional to $O(2^{|Pu|-1})$. The attainable envisioning through (step 2) to (step 4) is performed under the unspecified assumptions of $Pui=P-Pf \cup Psi$, and its initial situations are limited to the preceding envisionment. Accordingly, its complexity could be less than $O(2^{|Pui|-1})$. These derive the complexity of $O(2^{|Pu|-1})+O(\sum_{i=2}^{n} 2^{|Pui|-1})$ in case of the maximum for the history-oriented envisioning. As the number of parial slices in a partial history, n, is independent with the assumptions, and also each |Pui| is eaqual or less than |Pu|, the complexity of the history-oriented envisioning can be quite small comparing with the standard.

4 An Example

The basic performance of the proposed history-oriented envisioning has been evaluated through the application to the control of a steam generator commonly used in power plants for electricity generation. 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,

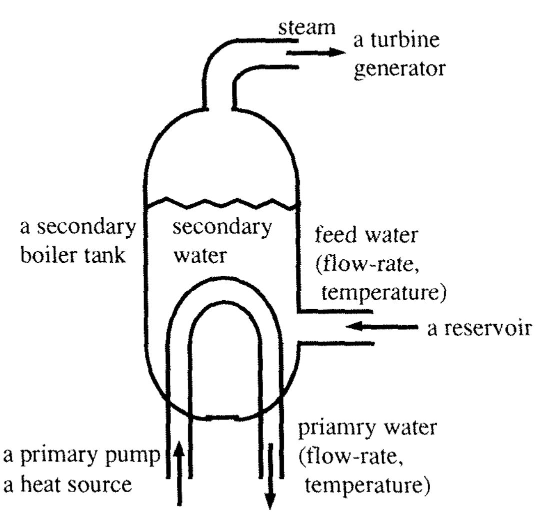

Fig. 5 A steam generator in a power plant for electricity generation.

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 history-oriented envisioning enumerates all possible plans within a finite number of envisionings.

A possible partial history corresponding to our mission can be written as 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 slices 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. 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 fixed portion of the scenario for this system and its partial history in fig.6 has been compiled to their assumptions and applied to the algorithm of history-oriented envisioning. A program specific to this type of examples has been developed and tested on a SPRAC-10 machine, while the development of a more generic program is currently under way. Figure 8 depicts the resultant envisionment indicating all possible and sound strategies to control the boiling under the given partial history. Totally, 29 situations were found. The author have tried to derive the total envisionment of this steam generator without specifying any partial history for comparison. However, the solution was not obtained due to the limitation of the memory capacity under the current program. The situations in the total envisionment can be at least more than 6000 even for this simple system, since it has 4 free boundary quantities. The planing of control strategies for process systems as this example is highly expensive, unless any constraints are introduced.

```
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.

5 Discussions and Related Works

One of the major characteristics of the history-oriented envisioning proposed here is the direct introduction of the specifications on the history of behaviors and actions to the envisioning process in addition to the conventional scenario. The envisioning focuses on only the specified situations and their

```
Partial Slice  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)] ?time)
                 <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
  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 Slice  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)] ?time)
                 =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)
  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.

histories, and derives those within small amount of computation. Iwasaki and Vescovi introduced a language named as CFRL to specify intended functional behaviors, and adopted it to a design support application [Iwasaki et al. 1993; Vescovi et al. 1993]. However, the CFRL just filters intended behaviors from the possible behaviors resulted in the envisioning, and hence does not control the envisioning process directly. In contrast, the characteristic of efficient behavior focusing of the history-oriented envisioning highly enhances the applicability of the envisioning theory to the practical scale problems.

Another major characteristic is the explicit use of the information on the behaviors' history we intend on the objective system not only our intentional actions to eliminate unrequired or useless solutions for our reasoning tasks. Drabble developed a system named as EXCALIBUR for planning and reasoning with process systems [Drabble 1993]. The system utilizes some attainable envisioning processes, and can manage the actions changing continuous process quantities not only the ones causing discontinuous change of views and processes. Also, it can take a tree and hierarchical structure of actions sequences. But, it does not handle the explicit specifications on the behaviors evolved in process systems in the envisioning. As many applications such as planning, simulation modeling and design in practical fields are usually seeking processes of objective or intended sequences of behaviors given in advance, the history-oriented envisioning provides an efficient approach to these synthetic tasks.

The third important characteristic is the incremental structure of the history-oriented envisioning. Some works on the measurements interpretation utilize the total envisionments of the objective system to interpret the situation transition of that system [Forbus 1986; DeCoste 1990, 1993]. The total envisioning which is quite expensive for practical scale processes is essential in their approaches. In contrast, the incremental feature of the history-oriented envisioning enables to take the information of a sequence of behaviors and actions one by one in the on-line monitoring process. If the amount of the input information is not small, then its attainable envisioning in each step will be quite cheap, and its real time use will be possible. Thus, this characteristic meets the practical needs of analytic tasks such as measurements interpretation, control and diagnosis.

6 Conclusion

A history-oriented envisioning method has been proposed in this work together with its basic input named partial slices and a partial history. The applicability and efficiency of this method has been readily confirmed through an example of a control strategy planning for a steam generator. The major characteristic of the history-oriented envisioning are summarized as follows.

(1) Soundness, small complexity and high efficiency comparing with the conventional envisioning.
(2) Envisioning focused on a sequence of intended partial behaviors and actions.
(3) Incremental envisioning to import the assumptions in an on-line manner.

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

Some following topics for our future works remains.

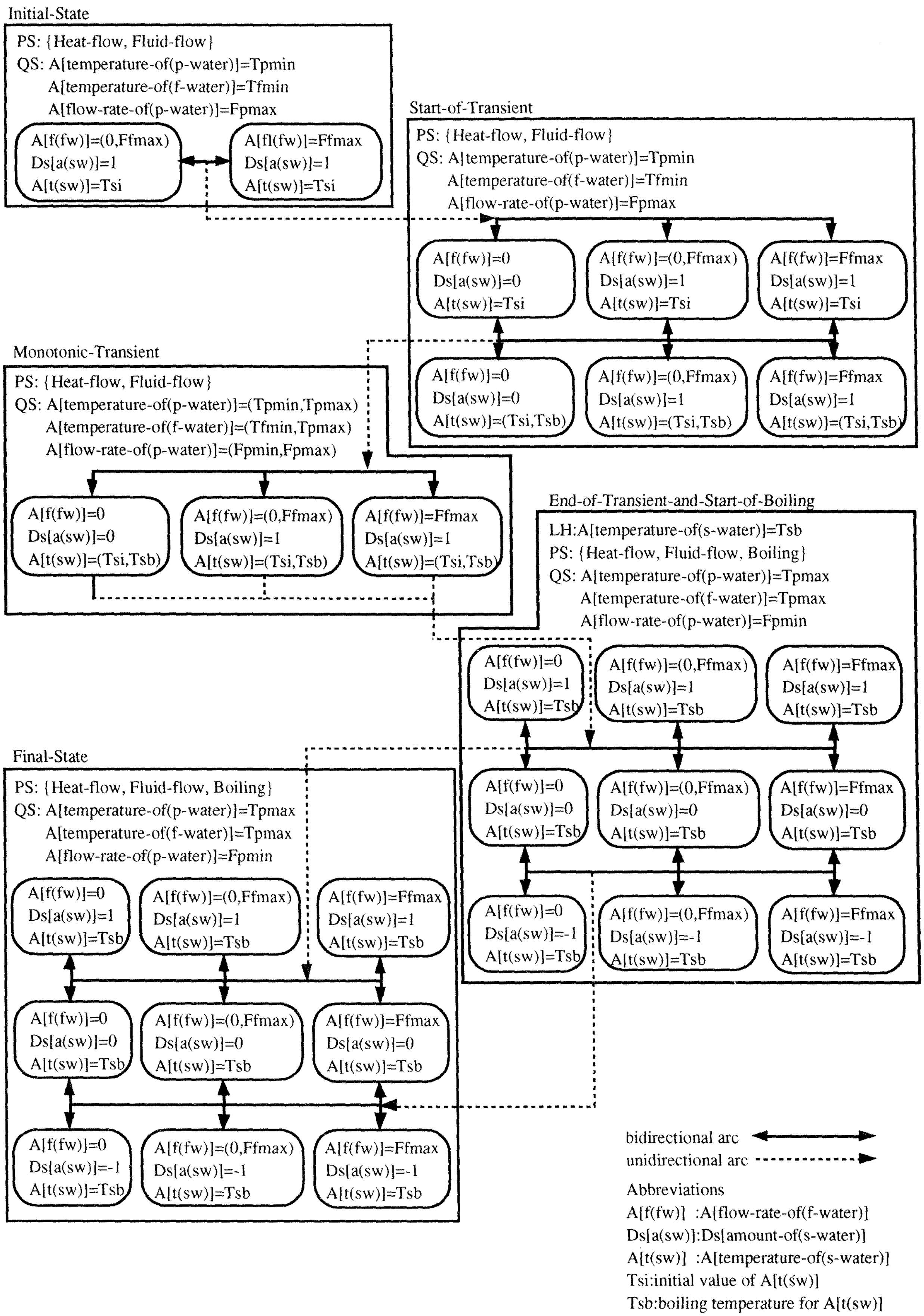

Fig.8 A situation transition diagram of a steam generator for a partial history.

(1) Extension of a partial history: The structure of a partial history is merely a sequence of partial slices, although the sequence can be fragmented by the "No-Specification" partial slice. Its extension to tree, graph and hierarchical structures or specified transition rules from a history likewise the EXCALIBUR and the CFRL will enhance the usability of the history-oriented envisioning.
(2) Seeking a better algorithm: The current algorithm for the history-oriented envisioning is a first version to evaluate the basic performance. Some more efficient algorithm might be developed.
(3) Development of a code for general use: The domain of process systems to be envisioned in the current program is quite limited. The author is currently working on the development of a general code for history-oriented envisioning.

Acknowledgments

The author wishes to express my thanks to Prof. Masaharu Kitamura in the Nuclear Engineering Department of Tohoku University for the useful discussions. The author extends the gratitude to Dr. Shuichi Koike and Dr. Hideaki Takahashi in Mitsubishi Research Institute, Inc. for their extensive support.

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.

[DeCoste 1990] Dennis DeCoste. Dynamic Across-Time Measurement Interpretation. In proceeding AAAI-90, pp.373-379, 1990.

[DeCoste 1993] Dennis DeCoste. Dynamic Across-Time Measurement Interpretation. Aritificial Intelligence, 51(1-3):273-341, 1993.

[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 1986] Kenneth D. Forbus. Interpreting Measurement of Physical Systems. In proceeding AAAI-86, pp.113-117, 1986.

[Forbus 1988] Kenneth D. Forbus. QPE: Using Assumption-based Truth Maintenance for Qualitative Simulation. Artifitial 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 Caulsality: Deriving Behavior from Structure. Aritificial Intelligence, 24:169-203, 1984.

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

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

[Ishida and Eshelman 1988] Y. Ishida and L. Eshelman. AQUA: Integrating Model-Based Diagnosis and Syndrome-Based Diagnosis, cmu-cs-87-111, 1987.

[Iwasaki et al. 1993] Yumi Iwasaki, Richard Fikes, Marcos Vescovi, and B. Chandrasekaran. How Things Are Intended to Work: Capturing Functional Knowledge in Device Design. In proceedings IJICAI-93, pp.1516-1522, 1993.

[Pearce 1988] D. A. Pearce. The Induction of Fault Diagnosis Systems from Qualitaive Models. In proceedings AAAI-88, pp. 353-357, 1988.

[Umeda et al. 1991] Yasushi Umeda, Tetsuo Tomiyama, and Hiroyuki Yoshikawa. A Design Methodology for a Self-maintenance Machine. DE-Vol.31, Design Theory and Methodology - DTM'91 - edited by L.A. Stauffer, Book No. G00641, 1991.

[Vescovi et al. 1993] Marcos Vescovi, Yumi Iwasaki, Richard Fikes, and B. Chandrasekaran. CFRL: A Language for Specifying the Causal Functionality of Engineered Devices. In proceedings AAAI-93, pp.626-633, 1993.

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

[Yannou 1993] Bernard Yannou. Qualitative Design with Envisionment. In working papers QR'93, pp.250-259. 1993.