

*Engineering*  
*Industrial & Management Engineering fields*

---

Okayama University

*Year 2006*

---

An interactive environment for  
generating sequential information

Mariko Sasakura  
Okayama University

Kenichi Iwata  
Okayama University

Susumu Yamasaki  
Okayama University

This paper is posted at eScholarship@OUDIR : Okayama University Digital Information Repository.

<http://escholarship.lib.okayama-u.ac.jp/industrial-engineering/4>

# An Interactive Environment for Generating Sequential Information

Mariko Sasakura, Kenichi Iwata, Susumu Yamasaki

Department of Computer Science,  
Graduate School of Natural Science and Technology, Okayama University  
{sasakura, iwa, yamasaki}@momo.it.okayama-u.ac.jp

## Abstract

*We propose an interactive environment in which we can generate sequential information. Sequential information is a sequence of movements which makes changes of situations. The system allows a user to construct an environment for generating sequential information. We provide interactions and visualization for sequential information generation based on theoretical grounds.*

**Keywords**—Sequential information generation, interactive system.

## 1 Introduction

Many movements in the world can be described by transitions of situations. For example, movements of a dance, or movements of a robot can be described as changes of places caused by movements of a body. For another example, movements of a melody of music can be described as changes of harmonies caused by sounds. The same mechanism can be found in the World Wide Web system.

In Web programming technology, a site page contains some knowledge as well as references to other pages, causing a knowledge state transition of observers. A sequence of pages observed may cause knowledge acquisition so that a sequence of knowledge-structure is meaningful.

We formulate such mechanism as a calculus with situations and charts which make changes of situations ([10]). We call sequences of charts and situations, which describe changes of situations, sequential information. In the case of Web programming, a page is regarded as an abstract chart, and a state of knowledge is interpreted as a situation. The chart involves a situation transition, where data and/or procedures may be described as contents of the chart. Such methodological concepts of chart and situation may be also captured in the case of training manual and its effect.

In this paper, we propose a system which provides an interactive environment for a calculus in order to generate sequential information. A graphical user interface can provide an easy way for a user to understand how the calculus forms sequential information. An interactive system

can provide an environment in which a user can modify conditions of the calculus easily. This system can support constructing a model of some application based on the calculus.

We determine the goal of the system as the following:

- The system can provide an interface for starting a query which asks sequential information under some conditions.
- The system can show sequential information as the result of the query which is a list of sequences of charts.
- The system can represent how the sequential information is produced.
- A user can change conditions and re-try the query immediately.

We have developed systems which visualize a reasoning process on logic programs ([8], [9]). The system proposed in this paper is an extended version of these systems for the sequential information. The system can be regarded as a kind of visual debugger of which usefulness is shown in [5], [7]. For Web browsing, there are some researches which visualize the history of browsing [1], [3], [4]. We have the same awareness of the problem as theirs. In this paper, we propose a new model for generating sequential information and an interactive system based on the model.

We introduce an example problem and represent a system design of the environment in Section 2. An implementation of a graphical user interface and interaction is described in Section 3. In Section 4, we describe the calculus which is a basis of the sequential information generation and show some properties of it. In Section 5, we show how the example is shown in the system.

## 2 A Design of the Interactive Environment for Generating Sequential Information

In this section, we give a design of the interactive environment for generating sequential information. At first, we mention the requirements for the system. Next, we introduce an example problem. Then, we describe the system

architecture. At last, we show the design of two screens: the main screen and the sequential information screen.

## 2.1 Requirements

We construct a system which generates sequential information under given conditions. Three conditions, namely, (1) a logic program, (2) a semantic function, and (3) constraints are given, and we represent them  $P$ ,  $Sem$ , and  $I$ , respectively in this paper.  $Sem$  specifies the relation between situations and charts: what chart can change situations.  $P$  specifies the relation between charts: what chart can be followed by other charts.  $I$  specifies the relation between situations: what situation can change to other situations. An input for the system is a pair of two situations: the initial and end situations. We call them as a query. An output of the system is sequential information which generates the changes of the given situations: the sequential information is a list of sequences of charts. Each sequence of charts represents a way to make the transition from the initial to the end situations. The sequential information represents all the candidates for changes of the initial situation to the end situation.

The system should satisfy the following requirements.

- A user can input the semantic function  $Sem$ , the logic program  $P$ , and the constraint set  $I$  to the system.
- A user can give a query to the system.
- The system can perform the transition of charts and generate sequential information.
- The system can show the result for the query.

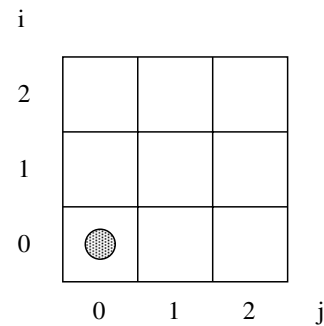
The detailed requirements about interaction are the following.

- A user can input a query which is a pair of the initial situation and the end situation.
- A user can know the result for a query that is more than one sequence of charts.
- A user can know how the result is derived. That is, the system can show how the result is obtained.

## 2.2 An example problem

We use an example problem for showing how the system generates sequential information. It is a variation of the example introduced in [6].

A counter starts at the left bottom square of the board in Figure 1, and may move within the board either *up* or *right*. The purpose of the original example in [6] is to describe movements of the counter in the formal way. In this paper, we give the initial position and final position to the counter, and find paths or sequences of movements of the counter. The answer for the question of how the counter moves from the left bottom square to the right up square



**Figure 1. An example problem: the initial situation in the board.**

can be shown by the interactive system for generating sequential information as in Figure 4. The details of the figure are mentioned in Section 5. We describe how we find them by the system.

## 2.3 System architecture

The system consists of

1. the calculus performer and
2. the interaction provider.

The calculus performer evaluates sequential information when a user inputs a query. It follows the procedure which we mention in Section 4.

The interaction provider prepares fields for a user's input and shows the result. It consists of

- the main screen, and
- the sequential information screen.

The main screen provides the fields for a user's input. It also shows  $P$ ,  $Sem$  and  $I$ . The sequential information screen shows a graph for selected sequences of charts. The details of the screens are discussed in the next subsections.

## 2.4 Main screen design

In the main screen a user can

- modify the logic program  $P$ , the semantic function  $Sem$  and the constraint  $I$ ,
- ask a query, and
- get sequential information for the query.

The main screen of the system has 6 fields of which names are Program, Semantics, Constraints, Start, Goal and Sequential Information.

Figure 3 shows the layout of the main screen design. Program shows  $P$ , Semantics shows  $Sem$  and Constraints shows  $I$ .  $P$ ,  $Sem$  and  $I$  are given to the system when it

is activated. A user can modify them at any time when the system is running. The top left field is prepared for editing  $P$ ,  $Sem$  or  $I$ .

Start and Goal are fields for a user input. When a user gives a query to the system, the user inputs the initial situation to the Start and the end situation to Goal.

Sequential information shows the sequences of charts for the user's query. It appears when the user puts the Go button. A sequence of charts is represented on a line. Six sequences of charts are shown in Figure 3.

## 2.5 Sequential information screen design

Figure 4 shows the sequential information screen design. It appears when a user puts the Show button in the main screen. The sequential information screen shows the details of selected sequences of charts.

In the screen, a sequence of charts is shown by situations and charts. A situation is represented as a rectangle which includes the name of the situation. A rectangle connects to other rectangles as an arrow if there is a chart which causes a situation transition between the two rectangles. The name of chart is drawn on the arrow.

If an arrow is clicked, the elements of Program, Semantics and Constraints (related to the situation transition) are coloured in the main screen. They show which elements of  $P$ ,  $Sem$  and  $I$  cause the situation transition.

If a user selects plural sequences of charts in the main screen and puts the Show button, the details of the sequences of charts are shown in the same sequential information screen.

By property shown in Section 4, we cannot avoid crossing edges in some cases [2]. We draw a graph in the sequential information screen as the following principles suggest.

- A situation is represented as a rectangle.
- A chart in a sequence is represented as an arrow. It connects two situations which are the before and after situations of the chart performed. The name of the chart is shown on the arrow.
- The initial situation is placed on the left end of a figure.
- The end situation is placed on the right end of the figure.

## 3 Implementation of Interaction

In this section, we discuss the interaction between a user and the system in the sequential information screen. We show how the interaction provider communicates with the calculus performer. We mention the theoretical grounds of the implementation of the interaction in Section 4.

### 3.1 Possible interactions in sequential information screen

In the sequential information screen, the following interaction needs communication for the calculus performer to:

1. Calculate sequential information after updating  $P$ ,  $Sem$  or  $I$ .
2. Change colour of the relative conditions when a user clicks an arrow in the sequential information screen.

The first item can be easily implemented as the calculus performer. To calculate sequential information makes use of conditions which are held by the interaction provider. Regarding the second item, we show the theoretical grounds for the interaction.

### 3.2 Communication between the calculus performer and the interaction provider

The communication between the calculus performer and the interaction provider occurs only at

- the start of a query and
- the end of the query.

At the start of a query, the interaction provider sends

- $P$ ,  $Sem$ , and  $I$
- the initial and end situations of the query

to the calculus performer.

At the end of the query, the calculus performer sends sequences of charts and situations with the first argument of each relation  $move_P$  to the interaction provider. The relation  $move_P$  is to construct the calculus for generating sequential information. We describe the details of  $move_P$  in Section 4.

When a user clicks an arrow in the sequential information screen, the interaction provider is responsible to change colours of the relative elements of  $P$ ,  $Sem$  and  $I$ .

## 4 A Calculus for Sequential Information Generation

### 4.1 A procedure of the calculus

The calculus [10] is regarded as a model of sequential information generation.

The calculus is

$$\mathfrak{S} = (C, \Sigma, Sem, P, I),$$

where  $C$  is a set of charts,  $C^{not} = \{not\ a | a \in C\}$  is a set of negative charts,  $\Sigma$  is a set of situations,  $Sem: C \rightarrow \Sigma \rightarrow \Sigma$  is a semantic function,  $P$  is a propositional logic program:  $P \subseteq C \times (C \cup C^{not})^*$ , and  $I \subseteq \Sigma \times \Sigma$  is a set of constraints.

- (1)  $\frac{}{move_P(\Box; \sigma; \sigma)}$
- (2)  $\frac{(A \leftarrow G) \in P \quad move_P(\leftarrow G; \sigma_1; Sem^{-1}[A]\sigma_2) \quad (Sem^{-1}[A]\sigma_2, \sigma_2)}{move_P(\leftarrow A; \sigma_1; \sigma_2)}$
- (3)  $\frac{failure_P(\leftarrow A)}{move_P(\leftarrow not A; \sigma; \sigma)}$
- (4)  $\frac{\text{No clause } A \leftarrow G \text{ with } A \text{ head is in } P}{failure_P(\leftarrow A)}$
- (5)  $\frac{\text{For all } A \leftarrow G \quad failure_P(\leftarrow G)}{failure_P(\leftarrow A)}$
- (6)  $\frac{move_P(\leftarrow A; \sigma_1; \sigma_2)}{failure_P(\leftarrow not A)}$
- (I)  $\frac{move_P(\leftarrow G_1; \sigma_1; \sigma_2) \quad move_P(\leftarrow G_2; \sigma_2; \sigma_3)}{move_P(\leftarrow G_1, G_2; \sigma_1; \sigma_3)}$
- (II)  $\frac{failure_P(\leftarrow G)}{failure_P(\leftarrow G_1, G, G_2)}$

**Figure 2. A procedure of the calculus.**

A chart is a cause of a situation transition. A situation represents a “situation” of a world. The details of a situation, that is how we describe a situation, is domain-specific.

An element of  $Sem$  is formulated as  $Sem[a]s_1 = s_2$  where  $a$  is a chart, and  $s_1$  and  $s_2$  are situations. It specifies a possible situation transition of which the initial situation is  $s_1$ , the final situation is  $s_2$ , and the chart which is a cause of the transition is  $a$ . The inverse function of  $Sem$  is  $Sem^{-1}[a]s_2 = s_1$  when  $Sem[a]s_1 = s_2$ .

$P$  represents possible chart sequences. Each literal in a clause of the logic program  $P$  is a chart. A clause  $l \leftarrow l_1, \dots, l_n$  ( $n \geq 0$ ) means the literal  $l$ , which is a chart or negative chart of the form  $not a$ , can be performed after the charts  $l_1, \dots, l_n$ .

$I$  specifies possible situation transitions. An element of  $I$  is denoted as  $(s_1, s_2)$  which means a situation  $s_1$  may change to  $s_2$  with some chart.

$Sem$ ,  $P$  and  $I$  can be specified individually. However, only the transition which satisfies all specifications of  $Sem$ ,  $P$  and  $I$  can be performed. We call  $Sem$ ,  $P$  and  $I$  *conditions* of the calculus.

We define two functions  $move_P$  and  $failure_P$ . The function  $move_P$  has three arguments: a sequence of charts, the initial situation and the end situation. It means the initial situation can change to the end situation by the charts. The function  $failure_P$  has an argument which is a sequence charts. It means the sequence of charts never occurs for the conditions.

The procedure of the calculus is described in formal way as in Figure 2. The query for the calculus, which asks whether some situation transition can be performed or not, can be solved by the procedure. The system receives the

initial situation and the end situation. It gets sequences of charts. We call sequences of charts and situations (which are changes before and after situations of each chart) sequential information by the procedure. The procedure finds the former situation from the end situation. This procedure finds all sequences of charts which change the initial situation to the end situation.

#### 4.2 Theoretical aspects for sequential information screen

In this subsection, we show two properties for making the sequential information screen.

**Property 1** *If we draw plural sequences of charts on a screen, they make a connected graph.*

**Proof** Sequences of charts have at least two same situations: the initial situation and the end situation. Then, two sequences of charts are at least connected at the initial situation and the end situation. A sequence of charts must be a connected graph because it represents transitions of situations. Therefore, if we draw plural sequences of charts on a screen, they make a connected graph. q.e.d.

**Property 2** *There are cases that a graph made by plural sequences of charts is not a plane graph.*

**Proof** There is a case that a graph made by plural sequences of charts may include a complete graph by 5 points:  $K_5$ . In the following case, the graph of sequences of charts from the situation  $s_0$  to  $s_4$  is a complete graph.

$$\mathfrak{S} = (C, \{s_0, s_1, s_2, s_3, s_4\}, Sem, P, I),$$

where:

$$\begin{aligned} C &= \{x_{(i,j)} | 0 \leq i \leq 4, 0 \leq j \leq 4, i \neq j\}, \\ Sem[x_{(i,j)}]s_i &= s_j \quad (i \neq j), \\ I &= \{(s_i, s_j) | i < j\}, \end{aligned}$$

and

$$\begin{aligned} P = \quad & x_{(0,1)} \leftarrow & x_{(2,3)} \leftarrow x_{(0,2)} \\ & x_{(0,2)} \leftarrow & x_{(2,3)} \leftarrow x_{(1,2)} \\ & x_{(0,3)} \leftarrow & x_{(2,4)} \leftarrow x_{(0,2)} \\ & x_{(0,4)} \leftarrow & x_{(2,4)} \leftarrow x_{(1,2)} \\ & x_{(1,2)} \leftarrow x_{(0,1)} & x_{(3,4)} \leftarrow x_{(0,3)} \\ & x_{(1,3)} \leftarrow x_{(0,1)} & x_{(3,4)} \leftarrow x_{(1,3)} \\ & x_{(1,4)} \leftarrow x_{(0,1)} & x_{(3,4)} \leftarrow x_{(2,3)}. \end{aligned}$$

q.e.d.

This property shows that we cannot avoid crossing edges in some cases [2].

### 4.3 Theoretical grounds for interaction

An arrow in the sequential information screen has at most an elements of  $P$ ,  $Sem$  and  $I$  as the relative condition. Because an arrow corresponds to one of the rule of the procedure in Figure 2. We find what information should be held in the interaction provider to show relations between a transition (represented as an arrow) and  $P$ ,  $Sem$  and  $I$ .

**Property 3** *An arrow with elements of the function  $Sem$  and the relation  $I$  can be constructed for given sequential information.*

**Proof** From the definition,  $Sem: C \rightarrow \Sigma \rightarrow \Sigma$  is a function and  $I$  is a subset of  $\Sigma \times \Sigma$ . Therefore, if a chart and two situations are given, we can find at most one element for  $Sem$  and  $I$  which make a situation transition with the given chart and situations. q.e.d.

**Property 4** *There is a case that an arrow with a clause of  $P$  cannot be uniquely constructed only for given sequential information.*

**Proof** In the case that a clause in  $P$  has a body which consists of more than two literals, we may not specify which clause has a relation to an arrow. For example, there is a calculus

$$\mathfrak{S} = (\{x, y\}, \{s_0, s_1\}, Sem, P, I)$$

where:

$$\begin{aligned} Sem[x]s_0 &= s_1, \\ Sem[y]s_1 &= s_1, \\ I &= \{(s_0, s_1), (s_1, s_1)\}, \end{aligned}$$

and

$$P = \begin{array}{l} x \leftarrow \\ y \leftarrow x. \end{array}$$

The situation transition from  $s_0$  to  $s_1$  can occur for both of  $x \leftarrow$  and  $y \leftarrow x$ . Therefore we cannot reconstruct the correspondence of an arrow and a clause of  $P$  only from sequential information. q.e.d.

Therefore, the system can make an arrow with its relative conditions by sequential information and the first argument of each  $move_P$ .

### 5 A Solution of the Example

Figure 3 shows the interface of the system for the example introduced in Section 2.2. We represent each square of the board as a situation, and  $up$  and  $right$  as charts. The chart  $u$  means  $up$  and  $r$  means  $right$ . The situation  $s_{ij}$  ( $0 \leq i \leq 2, 0 \leq j \leq 2$ ) represents the square of  $(i, j)$ .

The calculus  $\mathfrak{S} = (\{u, r\}, \Sigma, Sem, P, I)$ , where

$$\Sigma = \{s_{ij} \mid 0 \leq i \leq 2, 0 \leq j \leq 2\},$$

$P$  is:

$$\begin{array}{ll} u \leftarrow & r \leftarrow u \\ r \leftarrow & u \leftarrow u \\ r \leftarrow r & u \leftarrow r \end{array}$$

the function  $Sem$  assigns the transition of situations to each chart for  $0 \leq i \leq 2, 0 \leq j \leq 2$ :

$$\begin{aligned} Sem[u]s_{ij} &= s_{(i+1)j} \\ Sem[r]s_{ij} &= s_{i(j+1)} \end{aligned}$$

and

$$I = \{(s_{ij}, s_{(i+1)j}), (s_{ij}, s_{(i-1)j}), (s_{ij}, s_{i(j+1)}), (s_{ij}, s_{i(j-1)}) \mid 0 \leq i \leq 2, 0 \leq j \leq 2\}.$$

In the left side of Figure 3, the list named Program shows  $P$ , Semantics shows  $Sem$ , and Constraints shows  $I$ . The top of the field in the left side is for updating conditions.

In the right side of the figure, there are the fields named Start and Goal, Go and Show buttons and Sequential information list. When a user inputs an initial situation in Start and an end situation in Goal, and puts Go button, the calculus is started. Then the calculated sequential informations are represented in Sequential information list and Show button appears. Figure 3 shows the case in which a user inputs  $s_{00}$  in Start field and  $s_{22}$  in Goal field.

When a user puts Show button in Figure 3, Figure 4 appears. It shows the detailed information of the selected sequential information in Figure 3. The selected sequential information has the coloured background in the figure. In this case, two sequences of the six are selected. In the figure, the initial situation is placed in the left side and the final situation is placed in the right side. An arrow shows the change of situation according to the calculus. The characters on the arrow represents the chart for the change. The coloured elements in Program, Semantics and Constraints in main screen are related to the arrow labeled  $r$  from  $s_{01}$  to  $s_{02}$ .

### 6 Concluding Remarks

We develop an interactive system for generating sequential information. Making use of the system, a user can perform the calculus for sequential information generation, show its result, modify the conditions. The user can also perform the calculus under modified conditions in many times. The result of the calculus is shown by a graph. The feature of the system is that it has the theoretical grounds of the interaction. This paper is briefly concerned with the grounds.

The system can be applied to problems which have time dependent constraints and space dependent constraints. The time dependent constraints will be represented as a logic program  $P$  and the space dependent constraints will be represented as constraints  $I$ . We expect that the system can be used as the model of the WWW browsing.

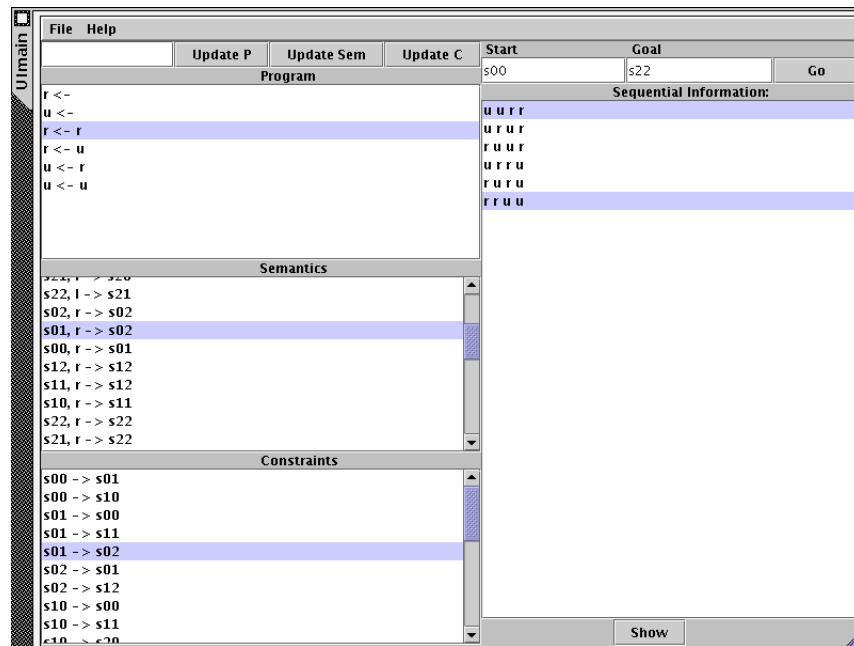


Figure 3. The user interface for the calculus.

## References

- [1] E. Ayers and J. Stasko, "Using graphic history in browsing the World Wide Web", *Proceedings of 4th International World Wide Web Conference*, pp.451–459, 1995.
- [2] G. D. Battista, P. Eades, R. Tamassia and I. G. Tollis. *Graph drawing*, Prentice Hall, 1999.
- [3] B. B. Bedreson, J. D. Hollan, J. Stewart, D. Rogers, D. Visk, L. T. Ring, E. Grose and C. Forsythe, "A zooming Web browser", *Human Factors in Web Development*, pp.255–266, 1998.
- [4] E. Frecon, "Webpath – a three dimensional Web history", *Proceedings of IEEE Information Visualization '98*, pp.3–10, 1998.
- [5] C. Geiger, W. Mueller and W. Rosenbach, "SAM – an animated 3D programming language", *Proceedings of 1998 IEEE Symposium on Visual Languages*, pp.228–235, 1998.
- [6] C.A.R. Hoare, *Communicating Sequential Processes*, Prentice-Hall, 1985.
- [7] T. Okamura, B. Shizuki and J. Tanaka, "Execution visualization and debugging in three-dimensional visual programming", *Proceedings of Eighth International Conference on Information Visualization (IV04)*, pp.167–172, 2004.
- [8] M. Sasakura, "Concentric circle diagrams for visualizing a reasoning process on an extended logic program", *PDPTA'02*, Volume I, pp.253–259, 2002.
- [9] M. Sasakura, and S. Yamasaki, "Visualization with hierarchically structured trees for an explanation reasoning system", *Proceedings of Eighth International Conference on Information Visualization (IV04)*, pp.893–898, 2004.
- [10] S. Yamasaki, and M. Sasakura, "A calculus effectively performing event formation with visualization", *Proc. of ISHPC 2005 (CD-ROM)*, 2005.

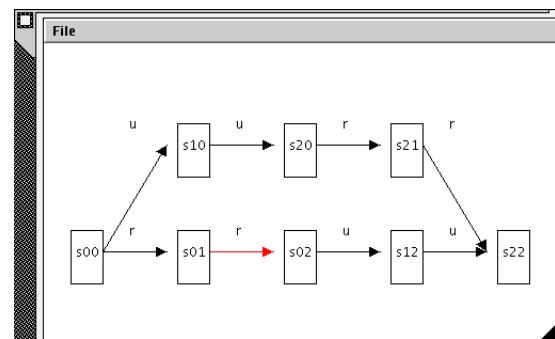


Figure 4. The sequential information screen.