INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'09
24 - 27 AUGUST 2009, STANFORD UNIVERSITY, STANFORD, CA, USA

DEVELOPMENT OF SCENE TRANSITION NETS(STN) GUI SIMULATOR FOR SERVICE FLOW SIMULATION

Takeshi Tateyama[1], Yoshiki Shimomura[1] and Seiichi Kawata[2]
(1) Tokyo Metropolitan Univ., Japan (2) Advanced Institute of Industrial Technology, Japan

ABSTRACT
Recently, a new academic field, "service engineering" has been very actively investigated. However, there are few effective software tools to simulate and evaluate services designed based on the concept of service engineering. In the past, the authors proposed a service flow simulation method using scene transition nets(STN) which is a graphic modeling and simulation method for discrete-continuous hybrid system. However, this method does not consider how to model and simulate complex service flows including interactions between agents in services. In this paper, the authors propose some extensions for service modeling and implement their functions to the STN GUI simulator. The experimental results of simulation of a DVD rental service shop including agents' interactions showed the availability of our method.

Keywords: scene transition nets(STN), Petri-nets, discrete-continuous hybrid systems, service engineering, simulation

1 INTRODUCTION
Recently, the non-good producing industries including retail trade, wholesale trade, and the service industries have been important because of their economical status. According to this background, a new academic field, "service engineering[1][2]" has been very actively investigated. The purpose of this research is to provide engineered methodologies of design and production of service. In service engineering, "service CAD[1]" called "service explorer" that is a software tool for service design is being proposed and studied. Designers can design services by using the service CAD based on engineered methodologies. However, there are few effective software tools to simulate and evaluate services designed based on the concept of service engineering. to overcome this problem, the authors has proposed a service flow simulation method[3] using Scene Transition Nets(STN)[4][5]. STN are very useful graphic modeling methods for discrete-continuous hybrid systems. They use concepts of "actors", "scenes" and "transitions" based on Petri nets[6] which is a modeling method for discrete event systems. By using this modeling and simulation method, designers can observe discrete flows of services designed by them and temporal changes in "degrees of satisfaction" of customers. In addition, the authors have developed "STN GUI simulator[3][7]". By using this GUI simulator, it is expected that designers can easily construct STN models and simulate them. However, this method does not consider how to simulate complex service flows including interactions between agents in services. The purpose of this study is to develop and improve the STN GUI Simulator in order to make it easy to construct STN models of more complex services and to simulate them. In this paper, the authors propose some extensions for service modeling and implement their functions to the STN GUI simulator. The experimental results of a service flow simulation of a rental DVD service shop including multiple agents(customers, DVD and clerks) shows the usefulness of our modeling method and software tool.

2 SCENE TRANSITION NETS(STN)
STN is a graphic modeling method for discrete-continuous hybrid systems: it uses a concept of "actors" and "scenes". It is based on Petri net which is a modeling method for discrete event systems. STN can express hybrid systems by using the concept of Petri net and by writing differential equations in the scenes. In STN, an actor corresponds to a subsystem of a hybrid system. Designers can simulate interactions between the sub systems that act in parallel by describing the models by using object-

oriented programming languages (e.g. Smalltalk, JAVA). STN is constructed by actors, scenes, transitions, and arcs shown as Figure 1. Details of these components are described below.

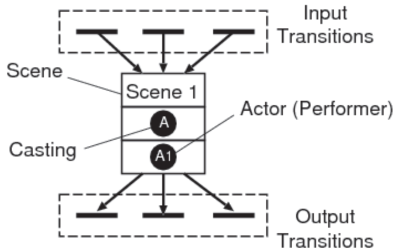


Figure 1. STN Components

2.1 Actor classes and actors

Actors in STN correspond to the tokens in Petri net. However, actors have state variables whose values change dynamically unlike tokens. An actor is one of the objects in STN that belongs to the "actor class". Actors belonging to the same actor class have common data structures(same types of constants and variables) and are called instances of the actor class. In STN, actors are defined as "subsystems" of a whole system, which is defined as an "observed system" and an observed system is a set of actors that interact with each other. Through these interactions, the states of actors and state variables change by the computing dynamics described in the scenes, as explained in the following section.

2.2 Scenes, castings and performers

Scenes in STN correspond to the places in Petri net. In STN, scenes are combinations of "activities" defined in discrete event systems and dynamics for changing variables of actors in the activities. Figure 1. shows STN components including a scene and an actor using the description format of STN. The circle A1 shown at the bottom of the scene indicates the location of actor named A1. The circle A at the middle of the scene indicates "casting" of the scene. A casting of a scene indicates an actor class whose instances(actors belonging to the actor class) can transit to the scene. In addition, an actor located in a scene is called a "performer" of the scene. Designers write dynamics using differential equations in each scene in order to dynamically change variables of the performers of the scene.

2.3 Transitions and arcs

"Transitions" in STN correspond to those in Petri net and indicate "scene transition boundaries" that correspond to "events" of discrete event systems. Transitions and scenes are connected by "arcs". Transitions connected to scenes with input arcs leading into the scenes are called "input transitions" of the scenes. In contrast, transitions connected to scenes with output arcs exiting from the scenes are called "output transitions" of the scenes. In a similar way, transitions have some "input scenes" and "output scenes". Designers write conditions for which the actors in the input scenes transit to the output scenes in the transitions and they write transit rules for the state variables.

3 STN GUI SIMULATOR

The authors have developed an STN GUI simulator so that designers can easily edit and simulate STN models. Figure 2. shows an overview of the GUI simulator written in the JAVA. This simulator consists of the STN edit toolbar, workspace, simulation toolbar, and graph windows. The details of these components are described below.

3.1 STN edit toolbar

The STN edit toolbar which is located at the top of the screen includes ten icon buttons. Designers can easily edit STN models by using this GUI tool.

EDIT AND SETUP BUTTONS FOR STN COMPONENTS

This toolbar includes eight edit buttons for STN components. By using these buttons, designers can easily construct STN in the workspace.

GLOBAL VARIABLE BUTTON
This button is used to set global variables. Designers can set the names, max/min values, and initial values of the variables and determine whether the simulator displays their graphs. In addition, they can set the dynamics by writing differential equations.

PROPERTY BUTTON
This button is used to set the following properties of STN objects.
(1) Actor class names and their variable names and attributes(integer or real)
(2) Actor names, their variables (max/min/initial values), and determining whether the simulator displays their graphs
(3) Scene names and dynamics of their performers
(4) Transition names, their fire conditions, and transit rules
(5) Capacities of castings and priorities of actors' transitions(First In First Out, FIFO / Last In First Out, LIFO)

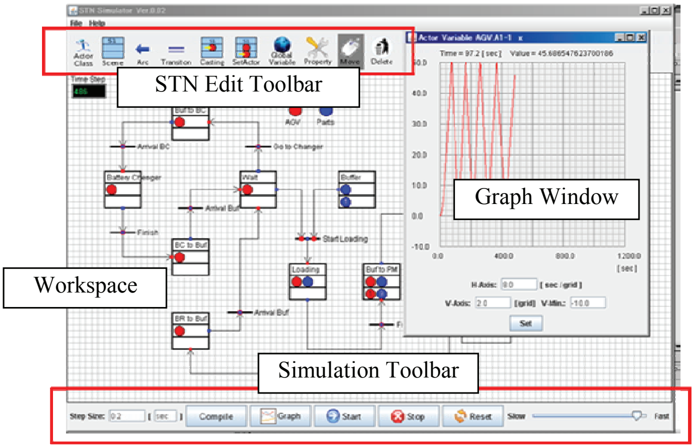


Figure 2. Overview of the STN GUI Simulator

3.2 Workspace

The workspace is the space in the middle of the simulator. Designers draw STN models in this space and can analyze behaviors of the networks by observing the animation in the space in simulation phases.

3.3 Simulation toolbar

In a simulation phase, designers control the simulation using the simulation toolbar. The designers can set the step size, time scale and the simulation speed and can start, stop and reset the simulation by using this toolbar. In addition, the designers execute to "compile" the constructed STN by using the compile button in this toolbar. When the compile button is clicked, the simulator analyzes the variables and equations(dynamics, fire conditions, and transit rules) inputted by the designers using a syntax analysis algorithm. The analysis algorithm will hereinafter be described in detail.

3.4 Syntax analysis algorithm for equations

The simulator includes a compiler to analyze the text data of the equations(dynamics of state variables, fire conditions, and transit rules of transitions) using a syntax analysis algorithm.

3.5 Sequence of the STN simulator

This section describes the sequence of STN simulation. First, the simulator searches the transitions which can fire in the network. In STN, a transition can fire only when there are performers of all casting in all input scenes of the transition and these performers' state variables meet the fire conditions of the transition. When some transitions fires, the actors which are performers of the input scenes of the transitions transit to the output scenes and their state variables are changed according to the transit rules of the transitions. Next, all global variables and all actors' state variables are updated using Runge-Kutta method. After that, simulation clock is updated. Above 1-step cycle is repeated until termination of the simulation.

3.6 Graph displays of actors' state variable and global variables

In the simulation phases, some graphs which show dynamical changes of designated actors' state variables and global variables. The designers can analyze the behaviors of continuous variable systems by observing these graphs and the behaviors of discrete event systems by observing the animation that show the transitions of the actors.

4 DISCRETE-CONTINUOUS HYBRID SIMULATION USING STN GUI SIMULATOR

STN can express and simulate complicated discrete-continuous hybrid systems. In previous studies, STN has been often applied to manufacturing systems, plant systems, and carrier systems, etc. For example, Figure 3. indicates a STN model of an automated parts transportation system using a battery-powered AGV[7]. In this model, the temporal changes of position of the AGV and the voltage of the battery are modeled as continuous systems using differential equations. The changes of the states of the AGV and the parts are modeled as discrete event systems(for example, the AGV has some states, "going to the battery changer", "going to the parts processing machine" and "waiting", etc.). By using the STN GUI simulator, designers can construct the STN models, execute simulation and observe the behaviors of the system.

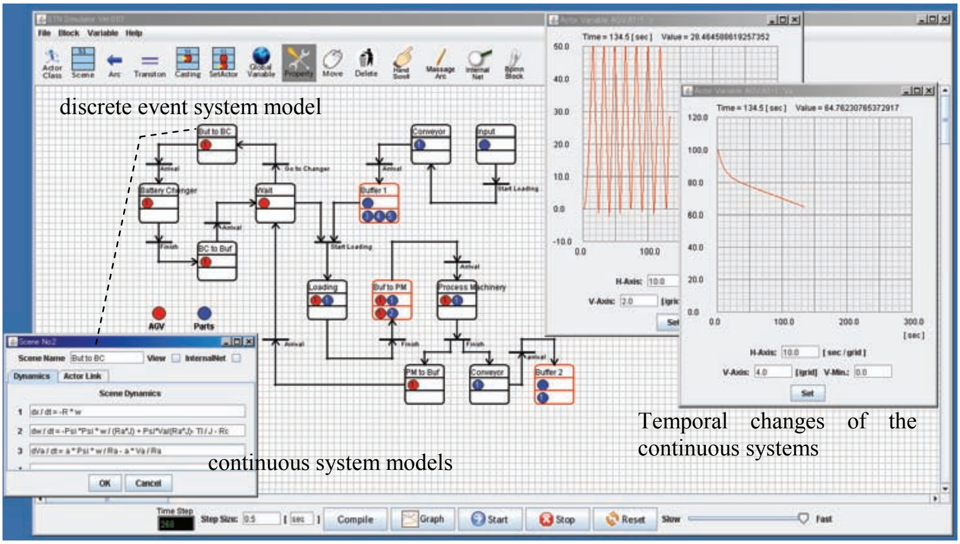


Figure 3. An Example of STN Simulation: An Automated Parts Transportation System Using A Battery-Powered AGV

5 SERVICE FLOW SIMULATION USING STN GUI SIMULATOR

In our study, services are regarded as complicated multi-agent and discrete-continuous hybrid systems. The authors propose to construct service flow models using STN and simulate the flows using STN GUI simulator in order to observe the agents' behaviors and evaluate the services.

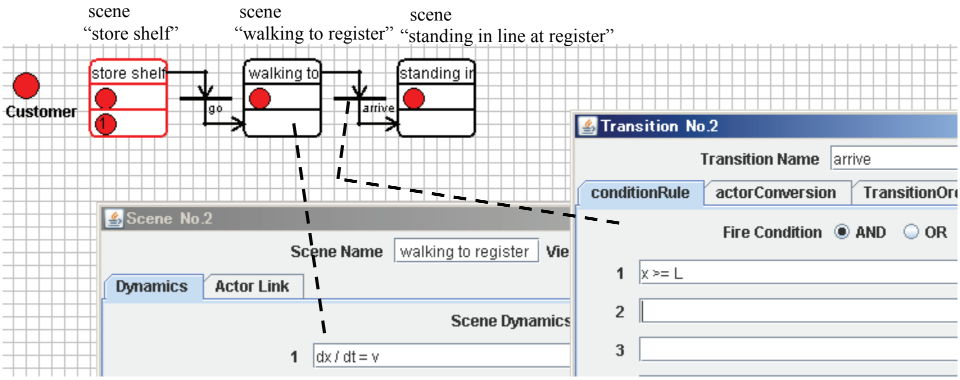


Figure 4. An Example of Setting Windows of a Dynamics of State Variable and a Transition Rule in the STN GUI Simulator

5.1 Service modeling

In STN service flow models the authors define, actors of a STN model correspond to agents of service providers and receivers in services, for example, customers as receivers and clerks as providers. Scenes correspond to service events.
A service flow model is constructed by the following procedure:

1. Write a "service script[8]" for the customers' behaviors from start to finish in the service.

2. Define some actor classes and actors as provider agents and receiver agents in a service. In addition, define the actors' state variables associated with the service.

3. Construct the customers' STN flow models which represent flows of the service the customers receive as discrete event systems according to the service script.

4. Specify some scenes of "service encounters"[8] in the customers' STN models. They are service events in which some service providers and customers directly interact.

5. Generate some scenes for the service providers that correspond to above service encounters.

6. Construct the service providers' STN flow models which include above scenes corresponding to the service encounters.

7. Construct one network by integrating the common service encounters' scenes of the customers and the providers.

8. Construct some dynamics models of the actors' state variables as continuous systems by using differential equations and write the equations in the scenes.

9. Define transition rules of the service events and write these rules in the transitions in the STN flow models.

For example, a customer walks L[m] from a store shelf to a register at the speed of v[m/s]. The customer has a state variable x[m] indicating his current position between them. In this case, the dynamics of the variable x is expressed in the following differential equation: $dx / dt = v$ and the transition rule to transit from an event "walking to register" to an event "standing in line at the register” is expressed in the following equation: $x >= L$. The setting windows of the STN GUI simulator are shown in Figure 4.

5.3 Expansion of STN and useful modeling techniques for service flow modeling

In this section, the authors extend STN in order to make it possible to construct complicated STN service flow models and show useful modeling techniques for them.

5.3.1 Definition of capacities of casting

In our STN researches, we have not defined a maximum number of actors which can transit to a scene at the same time. However, we need this definition when we construct STN models of complicated services and simulate them. For example, they are services which include a queue of cash registers or a service event limited number of customers can receive at the same time. Therefore, the authors propose that a casting has a parameter which is called "capacity". A capacity of a casting in a scene denotes the maximum number of the actors which can transit to the scene at the same time. Figure 5. shows the concept and outline of capacities of castings. In addition, when we simulate a queue, we need to specify transition rules of queued actors. In this study, we introduce a concept of two transition rules: FIFO(First In First Out) and LIFO(Last In First Out). Users can select one of the two transition rules in each scene. This new function enables us to simulate complicated systems including queues and complex transition rules.

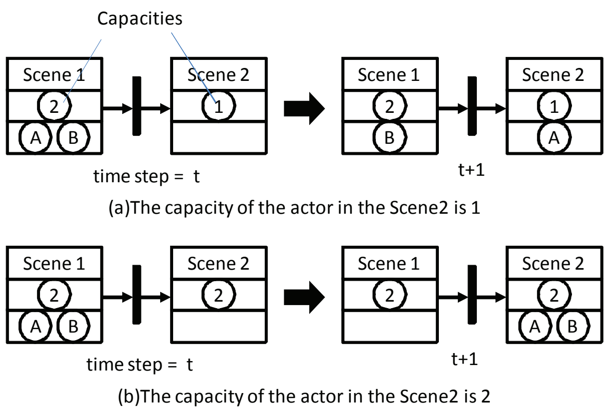


Figure 5. Concept of Capacities of Castings in Scenes

5.3.2 Definition of global variables

Flows of services and customers’ degree of satisfactions are affected by some their environmental elements nobody in the services can control. For example, these elements correspond to external temperature, seasons, economic conditions and penetration rate of Blu-ray Disc players, etc. When designers want to include these elements in the STN models of the services as parameters, it is necessary to use “global variables” and “global constants” which are common variables and constants in all scenes in STN. Because a dynamics written in a scene can’t refer and update state variables of actors in other scenes, the authors define the concept of “global variables” and implement new function to use them in the STN GUI simulator. Figure 6. shows the setting windows for defining global variables and their dynamics.

5.3.3 Description of stochastic elements

Services include many stochastic elements because they are severely affected by humans’ behaviors and interaction between humans. Therefore, it is necessary to include stochastic elements in the STN models of services in order to simulate them exactly. In the STN GUI simulator, designers can write a function which generates random real numbers in scenes. By using this function, designers can simulate flows of services including some stochastic elements(for example, action selections of actors using Boltzmann Distribution or roulette selection method, and generation of noises based on normal distribution, etc.) easily.

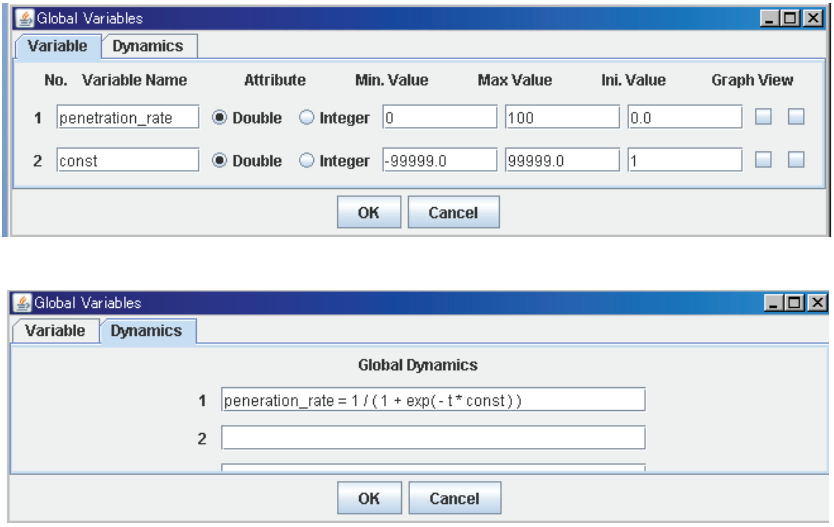


Figure 6. Setting Windows for Defining Global Variables(Upper) and Their Dynamics(Lower)

5.4 Customers' degree of satisfactions and their update rules

One of the purposes of design of services in service engineering is to design services which maximize "degree of satisfaction" of the service receivers. In this paper, the purpose of executing service flow simulation is to observe flows of the services, to compute degree of satisfaction of the service receivers and to evaluate these values. The degree of satisfaction is updated in each scene. The value of the degree of satisfaction of an actor a, S_a is computed using following equation:

$$S_a \leftarrow S_a + \sum_{i=1}^{N_n} DV_i^n LI_i^n \qquad (1)$$

Here, n is the current scene No. of the actor a, N_n is the number of "attribute parameters", AP which concern "receiver state parameters", RSP of the receivers in the scene n. A RSP represents a state of a receiver and an AP represents a parameter which concerns the change of the relevant RSPs. A degree of satisfaction is represented by a set of RSPs. For further details, the reader should refer to the papers about service engineering[1][2]. Chapter 6 describes how to use above equation concretely.

5.5 Simulation and evaluation

By executing the service flow simulation, designers can observe discrete flows of services including interactions between the actors and temporal changes in the degrees of satisfactions of the customers. By evaluating the results, the designers can improve the designing services.

6 EXAMPLE: DVD RENTAL SERVICE SHOP

This chapter describes an example of modeling and simulation of a DVD rental service shop. The outline of flow of the service is described in the following "service script" of a customer: (1) Leave home (2) Go to a rental shop (3) Enter the shop (4) Search DVD the customer wants to rent (5) Take the DVD (6) Go to the cash register (7) Stand in line at the cash register (8) Do order processing and pay money (9) Leave the shop (10) Go home (11) Reach home (12) Watch the movie (13) Leave home (14) Go to the shop for returning the DVD (15) Enter the shop (16) Go to the register (17) Return the DVD (18) Leave the shop (19) Go home (20) Reach home.

Next, we construct the STN model of the customer's flow based on the service script and specify service encounters. In this STN model, a service event "(8) order processing and pay money"

corresponds a service encounter. After construct the STN model of the flow of service providers including this encounter, construct one network by integrating the common service encounters' scenes of the customers and the providers. Figure 7. shows the STN model of the DVD rental service constructed according to above procedure. In this STN model, there are three actor classes: customer, DVD, and clerk. This network includes the flow of DVD and the flow of a provider who carries the returned DVD to the DVD rack. This network simulates flow of the customers who want to rent same DVD movies and flow of these DVD. The number of the DVD is limited and the customers who can't rent them go home without taking a DVD. In order to simulate queues of the cash registers, customers who rent other DVD movies come to the cash registers. The purpose of this modeling and simulation is to observe the actors' behaviors, compute the customers' degree of satisfaction and evaluate them. This service model includes following four design values. DV^1_I: distance from the customer's home to the shop((far)0.1 – (close)1.0), DV^2_I: average time required to find the DVD the customer want to rent((slow)0.1 – (fast)1.0), DV^3_I: waiting time at the cash registers, DV^4_I: price((high)0.1 – (low)1.0). The design value DV^1_I, DV^2_I and DV^4_I are arbitrarily determined by designers. The design value DV^3_I is computed by using results of simulation because this value is affected by interactions between some actors. This value is computed by the reciprocal of the waiting time. The customer's degree of satisfaction is computed by equation (1) defined in the section 5.4. In this experiment, the values of the level of importance are determined as follows: $LI^1_I = 0.4$, $LI^2_I = 0.2$, $LI^3_I = 0.2$, $LI^4_I = 0.2$.
Figure 8(a) shows the temporal changes of the degree of satisfaction S_a of a customer when the design values are as follows: $DV^1_I = DV^2_I = DV^4_I = 1.0$. And Figure 8(b) shows the S_a when the design values are as follows: $DV^1_I = DV^2_I = DV^4_I = 0.3$. These results indicate the effect of the design values on the degree of satisfactions. Table 1 shows the changes of average, maximum and minimum values of S_a when the number of the cash registers changes. These values are computed by multi-agent simulation including 10 customers. This result indicates the effect of the number of the cash registers and the waiting line on the degree of satisfaction. In addition, we can observe behaviors of the customers, DVD and clerks and changes of the waiting line by observing the animation provided by the STN GUI simulator.

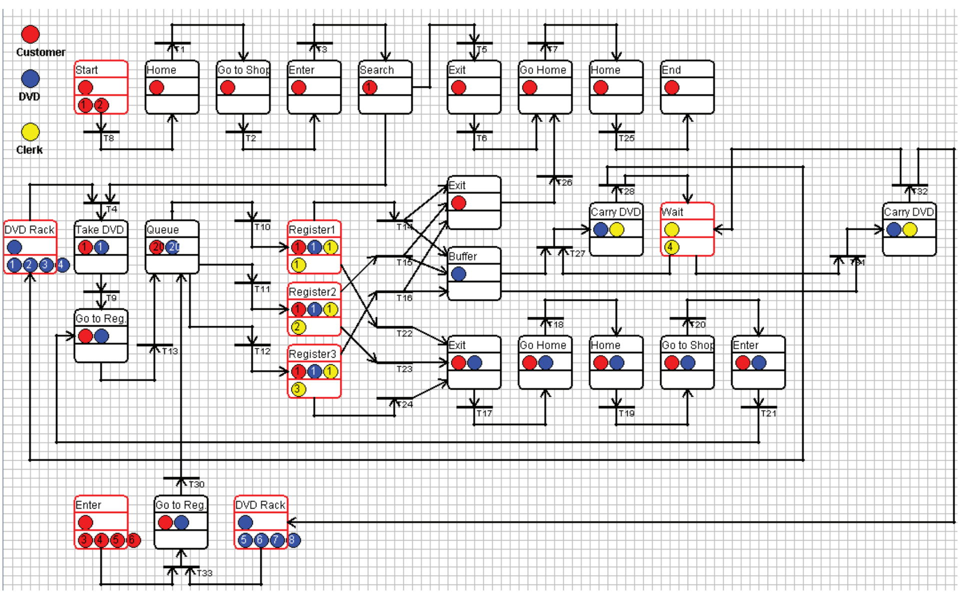
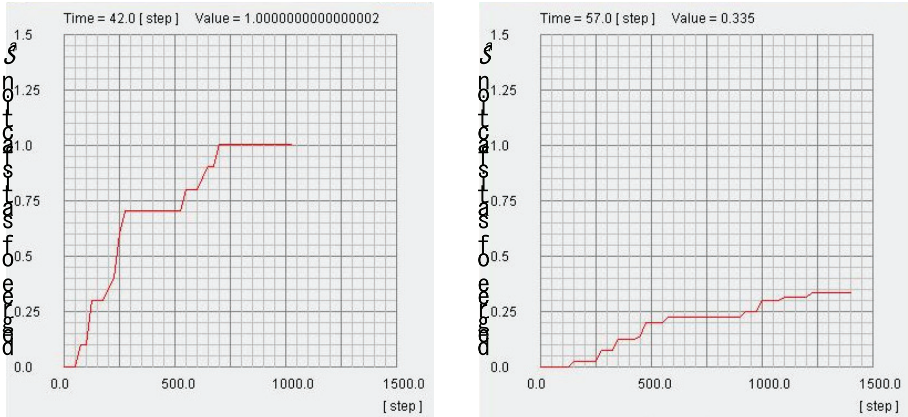


Figure 7. A STN Model of DVD Rental Service Shop



(a) Design Values: $DV^1_1 = DV^2_1 = DV^4_1 = 1.0$ *(b) Design Values*: $DV^1_1 = DV^2_1 = DV^4_1 = 0.3$

Figure 8. Temporal Changes of the values of the Customer's Degree of Satisfaction Sa

Table 1. Changes of the degree of satisfaction S_a when the number of cash registers changes

[redacted]

7 CONCLUSION

This paper described development and improvement of the STN GUI Simulator in order to make it easy to construct STN models of more complex services and to simulate them. The authors proposed some extensions for service modeling and implemented their functions to the STN GUI simulator. The experimental results of a service flow simulation of a rental DVD service shop including multiple agents showed the usefulness of our modeling method and software tool.

ACKNOWLEDGMENT

This research was partially supported by the Ministry of Education, Science, Sports, and Culture through a Grant-in-Aid for Scientific Research (B), 18360079

REFERENCES

[1] Shimomura Y. and Tomiyama T., Service Modelling for Service Engineering, *In Knowledge and Skill Chains in Engineering and Manufacturing: Information Infrastructure in the Era of Global Communications, IFIP International Federation for Information Processing, Vol. 168*, E. Arai, J. Goossenaerts, F. Kimura, K. Shirase (Eds.), Springer, ISBN-0387238514, 2005, pp. 31-38.

[2] Shimomura, Y., Sakao, T., Sundin, E. and Lindahl, M., A Design Process Model and a Computer Tool for Service Design, *In Proceedings of the 12th Design for Manufacturing and the Life Cycle Conference - DFMLC2007 -*, CD-ROM, The American Society for Mechanical Engineering (ASME), 2007.

[3] Tateyama, T., Kawata, S. and Shimomura, Y., Development of Scene Transition Net(STN) GUI Simulator for Discrete-Continuous Hybrid Systems, *In Proceedings of the 7th Japan Korea Workshop on CAD/CAM -Design Engineering Workshop-*, 2007, pp.82-87.

[4] Kawada, S., Kawata, S. and Watanabe, A., The discrete/continuous hybrid simulation using the scene transition net, *Proc. of the First Asian Control Conference(ASCC94), Vol.1*, 1994, pp.567-570.

[5] Kawada, S., Kawata, S. and Watanabe, A., A Manufacturing System Modeling Using a Scene Transition Net, *New Direction in Simulation for Manufacturing and Communication*, 1994,

pp.86-92.
[6] Murata T. Petri Nets: Properties, Analysis and Application. *Proceedings of the IEEE*, 77(4), 1989, 541-580.
[7] Tateyama, T., Chin, H, Shimomura, Y. and Kawata, S., Development and improvement of scene toransition nets(STN) GUI simulator for discrete-continuous hybrid systems, *Proceedings of the 4th China-Korea Joint Conference on Geometric and Visual Computing & IJCC Workshop (CKJC2008 & IJCC Workshop)*, 2008, pp.140 – 143.
[8] Fisk, R.P., Grove, S. J, and John, J., *INTERACTIVE SERVICE MARKETING 2nd edition*, Houghton Mifflin Company, 2004.

Contact: Takeshi Tateyama
Tokyo Metropolitan University
Graduate School of System Design
Asahigaoka 6-6
191-0065, Hino-shi, Tokyo
Japan
Phone: +81-42-585-8472
Fax: +81-42-585-8472
E-mail: tateyama@sd.tmu.ac.jp

Researcher T. Tateyama is a research associate in the graduate school of system design, department of human mechatronics systems at Tokyo Metropolitan University.
He is interested in machine learning and service engineering.