A Novel Method based on Behavior Network for Web Service Composition

Myung-Chul Jung and Sung-Bae Cho
Department of Computer Science
Yonsei University
134 Shinchon-dong, Sudaemoon-ku, Seoul 120-749, Korea
mcjung@sclab.yonsei.ac.kr, sbcho@cs.yonsei.ac.kr

Abstract

It is difficult to produce the user's needed services by appropriately processing a number of the web resources. Web services enable to produce the relevant information to users via standard methods and to make more flexible service sequences using Web service composition. As a clerk helps us to buy items in the store, Web services should provide more active services in accordance with user's situation and environments. But the composition based on static rules is currently dominant in Web service composition. In this paper, we propose Web service composition based on behavior network. Behavior network generates behaviors automatically through the internal links and external links with sensors and goals, and can be applied to actively construct dynamic service sequences using Web services. To verify the usefulness of our the proposed method, we have developed an application based on behavior network using Amazon Web services. We have conducted the experiments with the application and have evaluated the usefulness. The results show the possibility of the behavior network for Web service composition.

1. Introduction

It has become increasingly difficult for users to find information on the Web that satisfies their individual needs since information resources on the web continue to grow. There are many studies about dynamic service composition for resolving this difficulty [1]. In this paper, we propose a dynamic Web service composition based on behavior network that constructs the relevant services for users by combining and reorganizing various information provided from Web services.

We make use of Web services to easily access and manipulate the resources on the Web via a method like remote procedure call. HTML documents have many problems in that grammar is not clear, the location of information is not fixed, and so on.

Web services are programmable application logic accessible using standard Internet protocols [2]. In terms of component-based development, Web services represent functionality that can be easily reused without knowing how the service is implemented. Web services are accessed via standard protocols such as HTTP and XML as a formatting language. There are essentially four key components of Web services[1] [3]. The fundamental building block is XML that is the data format to describe their interfaces and to encode their messages. The XML based language for describing Web services is called WSDL (Web Services Description Language). SOAP (Simple Object Access Protocol) is the protocol by which data is transported between a service requester and a service provider. Finally, there is UDDI [2] (Universal Description and Discovery of Information) that is a registry where available services are published.

A Web service represents a discrete unit of process, application, or system functionality. Each discrete Web service can be deployed on and accessed from any node on the Internet. Multiple services can be combined or assembled to deliver more valuable services. Recently, many studies about composition of Web services are in progress [4, 5].

There are many action selection mechanisms to combine behaviors for generating high-level behaviors [6, 7]. Behavior network, one of action selection mechanisms, can contain goals of the model in implicit manner and propagate the activation of behavior in two directions (forward and backward) through the network for dynamic selection, while most of action selection mechanisms cannot do [8, 9, 10]. Behavior network can be applied to control a mobile robot [11].

[1] Web services activity, http://www.w3.org/2002/ws
[2] http://www.uddi.org

Proceedings of the International Conference on Next Generation Web Services Practices (NWeSP'05)
0-7695-2452-4/05 $20.00 © 2005 IEEE

Behavior network can be good choice in dynamic Web services composition.

The system overview of Web services composition in this paper is shown in Figure 1. It describes the generation of many action sequences selected by behavior network made from ontology about user's environments and actions.

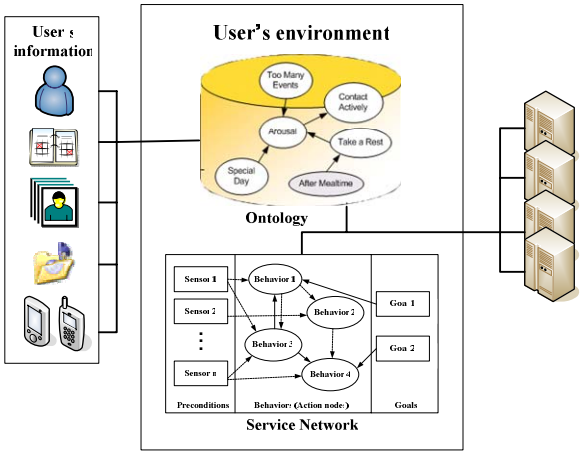

Figure 1: System overview of Web service composition.

2. Service Construction Mechanism

For dynamic Web service composition, we make service nodes and compose them properly according to user's environment. The method of action selection is an essential element to arbitrate activation of the node for the global goal. Behavior network that is one method of action selection can represent the action order based on links between nodes.

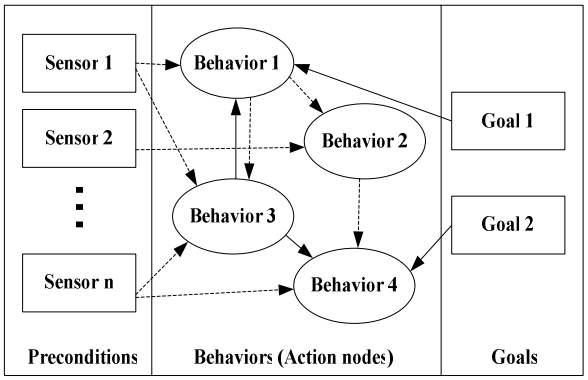

Figure 2: An example of Behavior Network. Solid lines denote goal or predecessor links, and dashed lines denote sensor or successor links.

Table 1: Internal links

Predecessor link	If proposition X is false, proposition X is a precondition of node A, and proposition X is in the add list of node B, then there is an active predecessor link from A to B
Successor link	If proposition X is false, proposition X is in the add list of node A, proposition X is a precondition of node B, and the node A is executable, then there is an active successor link from A to B
Conflictor link	If proposition X is true, proposition X is a precondition of node A, And proposition X is in the delete list of node B, then there is an active conflictor link from A to B

Table 2: External links

From sensors of the environment	If proposition X about the environment is true, and proposition X is a precondition of node A, then there is an active link from the sensor of the proposition X to node A
From goals	If goal Y has an activation greater than zero, and goal Y is in the add list of node A, then there is an active link from the goal Y to node A
From protected goals	If goal Y has an activation greater than zero, and goal Y is in the delete list of node A, then there is an active link from the goal Y to node A

2.1 Behavior Network

A behavior network consists of action nodes, internal links, and external links and is shown in Figure 2. Each action node is composed of preconditions, add list, delete list, activation level, and the running code. The preconditions are a set of logical conditions about the environment that are required to be true in order to execute the action of the node. The add list consists of conditions about the environment that may be true when this node is executable. The delete list consists of conditions that are likely to be made false by the execution of this node. The activation level means a degree that this node can be executable. And the last component, the running code put through a program code if the node is executed. The internal links are specified in Table 1. The external links providing input to the network are specified in Table 2. Table 1 and 2 describe the links in the behavior network.

2.2 Control of Behavior Network

Competition of behaviors is the basic characteristics of behavior network. Each behavior attempts to get higher activation level than other behaviors from activation spreading in forward and backward directions. Among candidate behaviors, one that has the highest activation level is selected and has a control. Table 3 describes what each parameter means. Activation level a of behavior is calculated as follows.

Table 3: Five parameters are used to configure behavior network

π	The mean level of activation.
θ	The threshold of activation, where θ is lowered 10 % every time no module could be selected, and is reset to its initial value whenever a module becomes active.
ϕ	The amount of activation energy injected by the state per true proposition.
γ	The amount of activation energy injected by the goals per goal.
δ	The amount of activation energy taken away by the protected goals per protected goal.

Forward propagation: Activation a is updated as the value added by environmental sensors that are precondition of the behavior. Precondition is the sensor that is likely to be true when the behavior is executed. n means the number of sensors. a_s is the activation level of the sensor.

$$\Delta a_1 = \sum_{i=1}^{n} f(a_{s_i})$$

$$f(a_{s_i}) = \begin{cases} \phi \times a_{s_i}, & s_i \in \text{precondition} \\ 0, & s_i \notin \text{precondition} \end{cases}$$

Backward propagation: Activation a is updated as the value added by goals that are directly connected to the behavior. If the execution of the behavior is desirable for the goal, positive goal-behavior link is activated. Otherwise, negative goal-behavior link is activated. n means the number of goals. a_g is the activation level of the goal.

$$\Delta a_2 = \sum_{i=1}^{n} f(a_{g_i})$$

$$f(a_{g_i}) = \begin{cases} \gamma \times a_{g_i}, & g_i \in \text{positive link} \\ -\delta \times a_{g_i}, & g_i \in \text{negative link} \end{cases}$$

Internal spreading: activation a is updated as the value added by other behaviors that are directly connected. If the execution of behavior B is desirable for behavior A, predecessor link from A to B and successor link from B to A are active. If the execution of behavior B is not desirable for behavior A, conflictor link from A to B is active. Here, n is the number of behaviors, and a_b is the activation level of the behavior.

$$\Delta a_3 = \sum_{i=1}^{n} f(a_{b_i})$$

$$f(a_{b_i}) = \begin{cases} a_{b_i}, & \text{predecessor link from } b_i \\ \gamma \times a_{b_i}, & \text{successor link from } b_i \\ -\delta \times a_{b_i}, & \text{conflictor link from } b_i \\ 0, & \text{otherwise} \end{cases}$$

Finally, the activation of a is updated as follows.

$$a' = a + \Delta a_1 + \Delta a_2 + \Delta a_3$$

If the activation level a' is larger than threshold Θ and precondition of the behavior is true, the behavior becomes candidate to be selected. Among candidate behaviors, the highest activation behavior is selected for execution, threshold Θ is reduced by 10% and the activation update procedure is repeated until there are candidate behaviors.

2.3 Procedure of Action Selection

Action selection procedure of an activation network is as follows.

1. Calculate the excitation coming in from the environment and the motivations.
2. Spread excitation along the predecessor, successor, and conflictor links.
3. Normalize the node activations so that the average activation becomes equal to the constant π .
4. Check to see whether any nodes are executable and, if so, choose the one with the highest activation, execute it, and finish.
5. If no node is executable, reduce the global threshold and repeat the cycle.

3. Experimental Results

We have made the system using Amazon Web services[3] that support the proposed model. We evaluate the action pattern of the service agent while users buy the books on Amazon Web site[4].

Figure 4: Web service composition system.

Figure 4 shows the web service composition system. The top of the Figure 4 has the Web address input window and the user feedback slider. The center of that displays the Web that users surf. The bottom of that offers the services selected by the proposed method. The service window at the bottom is separated from the main window in order to minimize users' inconvenience since users can think that the services are dispensable.

3.1 Services

In this paper, 8 behaviors are defined. They consist of "Search," "Show history," "Show bestseller," "Collaborative recommendation," "Add to cart," "Show cart," "Show review," and "Nothing to do."

"Search" shows the search window so that the user can find the desired items. "Show history" shows the items watched by user. "Show bestseller" shows the summarized items that other buyers prefer to. "Collaborative recommendation" shows the favorite items by the buyers that chose the current item. "Add to cart" ask the user about purchasing the item. "Show cart" shows the items in the user's cart. "Show review" shows the review that is written by other buyers or experts.

[3] http://www.amazon.com/webservices
[4] http://www.amazon.com

3.2 Action Selection Model

In this section, we apply the action selection mechanism so as to construct Web services composition. Our environment requires 10 stages, such as "Bestseller interest," "Search interest," "History interest," "Similar items interest," "Current item interest," "Item," "Item satisfaction," "User action," "No action in limited time," and "Items in the cart." They are set as follows.

- "Bestseller interest" : the amount of user's interest in bestselling items.
- "Search interest" : the amount of user's interest in search for new items.
- "History interest" : the amount of user's interest in user's history that user watched.
- "Similar items interest" : the amount of user's interest in collaborative filtering recommendation items.
- "Current item interest" : the amount of user's interest in the item that the user watches now.
- "Item" : whether the current window displays an item.
- "Item satisfaction" : the user's feedback of items that is provided by the service agent.
- "User action" : whether a user is active now.
- "No action in limited time" : whether a user does not act in limited time.
- "Items in the cart" : whether the items exist in the online cart.

We set a goal such as "Buy." The final goal of the proposed system is that the user processes to purchase the items in the cart.

Table 4 describes preconditions of the nodes. The relationships among nodes are decided by successor links or predecessor links. If predecessor link is from A node to B node, successor link from B node to A node exists. If node A and node B are connected, they exchange their activation values while activation spreads. Table 5 describes add lists of nodes.

Figure 5 shows the behavior network for Web service composition. The dashed lines from sensors to nodes are external links about preconditions of nodes. The solid lines from the goal to nodes are external links about the goal. The dashed lines between nodes denote the predecessor links. The solid lines between nodes denote the successor links.

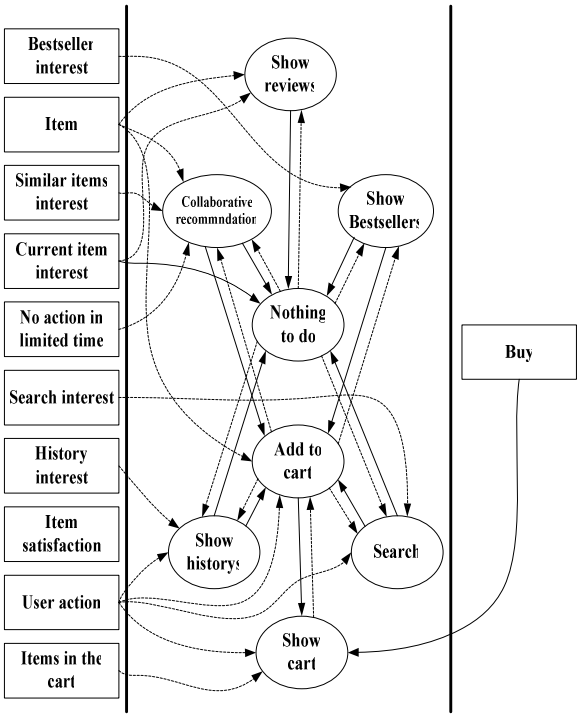

Figure 5: Behavior network for Web service composition. Solid lines denote goal or predecessor links, and dashed lines denote sensor or successor links.

Table 4: Preconditions of nodes

Node	Preconditions
Search	Search interest, User action
Show history	History interest, User action
Show bestsellers	Bestseller interest
Collaborative recommendation	Similar items interest, Item, No action in limited time
Add to cart	Item, User action
Show cart	User action, Items in the cart
Show reviews	Current item interest Item
Nothing to do	Current item interest

Table 5: Add lists of nodes

Node	Add lists
Search	Item, User action
Show history	Item, User action
Show bestsellers	Item, User action, Item satisfaction
Collaborative recommendation	Item, User action, Item satisfaction
Add to cart	Items in the cart
Show cart	Buy
Show reviews	Current item interest, Similar items interest
Nothing to do	No action in limited time

3.3 Results

We use the developed application for Web service composition to evaluate the usefulness of the proposed action selection model.

Figure 6 shows the action sequence while the user bought books using our application. The application selects a behavior at each time and executes the action node to achieve the goal. The number of action selections of the application is 134 times in total. In the early stage, "Search" is mainly selected to assist the user to find his favorites. "Show reviews," "Collaborative recommendation," and "Show Bestsellers" is fairly chosen from the 10th selection to the 40th selection while the user looks at the items. The user adds the item to the online cart between the 40th and the 50th selection. After that, the number of the action selection of "Show history" and "show cart" is increased. The user buys the items at the88th and the 125th selection and completes the purchase.

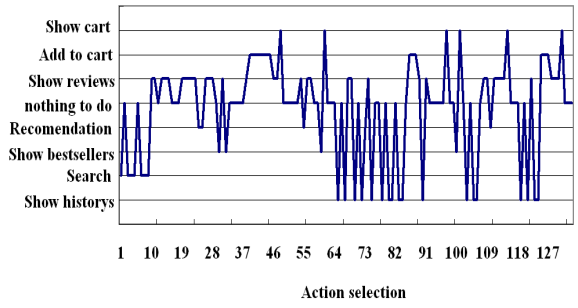

Figure 6: Action selection sequence of the Web services composition system

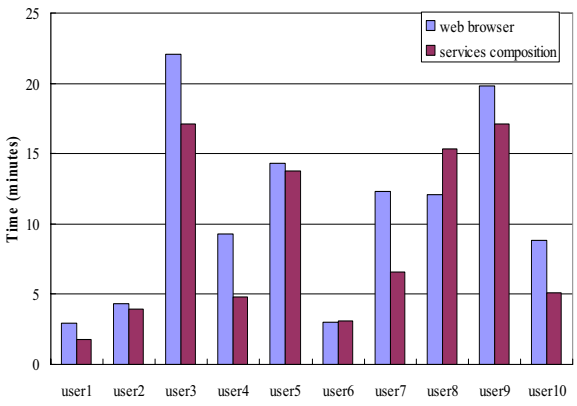

Figure 7: User test comparing normal web browser with Web service composition system

4. Conclusions and Future Works

In this paper, in order to provide each user with more relevant information, we have developed a Web service composition system based on behavior network dynamically. Our approach is novel in that it allows each user to construct a flexible composition, which is not performed well in the conventional Web service composition systems, by capturing environmental changes from sensors. In spite of the situation that is not predicted, a relevant solution can be obtained by action selection mechanism.

The application of the proposed model has been developed based on Amazon Web services. We have conducted experiments so as to verify the effectiveness of the approach. The experiments of an action selection analysis and user test have been performed. Our approach has constructed more appropriate behavior sequences to a user. It has just disadvantage in that the services provided from Amazon Web is similar to that provided from Amazon Web services because the application is developed based on Amazon Web services. But this problem is naturally solved by extending the applied domain of Web services.

This paper focuses on automatic composition of Web services. We show the usefulness of our approach to settle Web services composition. In the future, Web services composition based on UDDI that can find the more Web services will be investigated. If broadband networks spread widely, our general approach can be adapted to the services composition under ubiquitous environments.

Acknowledgements

This research was supported by Brain Science and Engineering Research Program sponsored by Korean Ministry of Commerce, Industry and Energy.

References

[1] J. J. Bryson, D. L. Martin, S. A. McIlraith and L. A. Stein, "Toward behavioral intelligence in the semantic web," *IEEE Computer*, vol. 35, no. 11, pp. 48-54, 2002.

[2] K. Gottschalk, S. Graham, H. Kreger, J. Snell, "Introduction to Web services architecture," IBM System Journal, vol. 41, no. 2, pp. 170-177, 2002.

[3] F. Curbera, M. Duftler, R. Khalaf, W. Nagy, N. Mukhi, and S. Weerawarana, "Unraveling the Web services Web: An introduction to SOAP, WSDL, and UDDI," *IEEE Internet Computing*, vol.6, no. 2, pp. 86-93, 2002.

[4] J. Rao and X. Su, "A survey of automated Web service composition methods," *In Proceedings of First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004)*, USA, July 2004.

[5] N. Milanovic and M. Malek, "Current solutions for Web service composition," *IEEE Internet Computing*, vol. 8, No. 6, pp. 51-59, 2004.

[6] P. Pirjanian, "Behavior coordination mechanisms state-of-the-art," *Tech-report IRIS-99-975*, Institute for Robotics and Intelligent Systems, School of University of Southern California, Oct 1999.

[7] M. Humphrys, *Action selection methods using reinforcement learning*, PhD thesis, University of Cambridge, Computer Laboratory, June 1997.

[8] P. Maes, "How to do the right thing," *Connection Science Journal*, vol. 1, no. 3, pp. 291-323, 1989.

[9] P. Maes, "Situated agents can have goals," *Journal for Robotics and Autonomous Systems*, vol. 6, pp. 49-70, 1990.

[10] T. Tyrrell, "An evaluation of Maes's bottom-up mechanism for behavior selection," *Adaptive Behavior*, vol. 2, pp. 307-348, 1994.

[11] K-J. Kim and S.-B Cho, "Robot action selection for higher behaviors with CAM-Brain modules," *In Proceedings of the 32nd International Symposium on Robotics*, pp. 1623-1628, 2001.