A Multi-agent Based Intelligent WWW Interfacer

Yasuhiko KITAMURA
Faculty of Engineering, Osaka City University
3-3-138, Sugimoto, Sumiyoshi-ku, Osaka, 558-8585, JAPAN
Phone & Fax: +81-6-6605-3081, E-mail: kitamura@info.eng.osaka-cu.ac.jp

Introduction

The WWW technology is rapidly proliferating into our society and, as an infrastructure that supports our daily life, it is widely used for various purposes such as e-commerce, research and education, personal or group information dissemination, special interest community creation and so on. The amount of WWW information increases very rapidly day by day, but that makes retrieving information more and more. Search engines are most widely used tool to retrieve information from the WWW, but it is not always very useful for novice users such as aged people. For example, when a user wants to know a recipe for pork, he/she may submit just "pork" as a keyword to a search engine, then he/she may get stuck with a large number of URLs including not only about recipes but also about farming, retailers, restaurants, and so on.

Generally speaking, there are two ambiguities concerning WWW information retrieval by using a search engine. The first ambiguity is about information sources. Currently the HTML is mainly used to describe WWW pages, but it is not well equipped for specifying semantic information in the pages. Hence, a standard search engine hits a page just when the page includes the specified keywords. The second ambiguity is about user's queries. For example, when a user submits a keyword "pork", he/she may have an intention like "I would like to cook a dish with pork. Yesterday I cooked a Chinese dish, so today a Japanese one. I am on diet now, so I prefer a low-calorie one." in his/her background. Of course, current search engines cannot accept the above intention as it is, they just can accept a few keywords that include in the intention.

To deal with the above ambiguities, we are currently developing a multi-agent based intelligent WWW interfacer as shown in Figure 1. Our system consists of multiple information agents, each of which provides domain specific information, and a personal agent, which manages its user's profile information. Each agent has its own character interface and an information retrieval task is represented through dialogues between the user and the personal agent and between the personal agent and information agents.

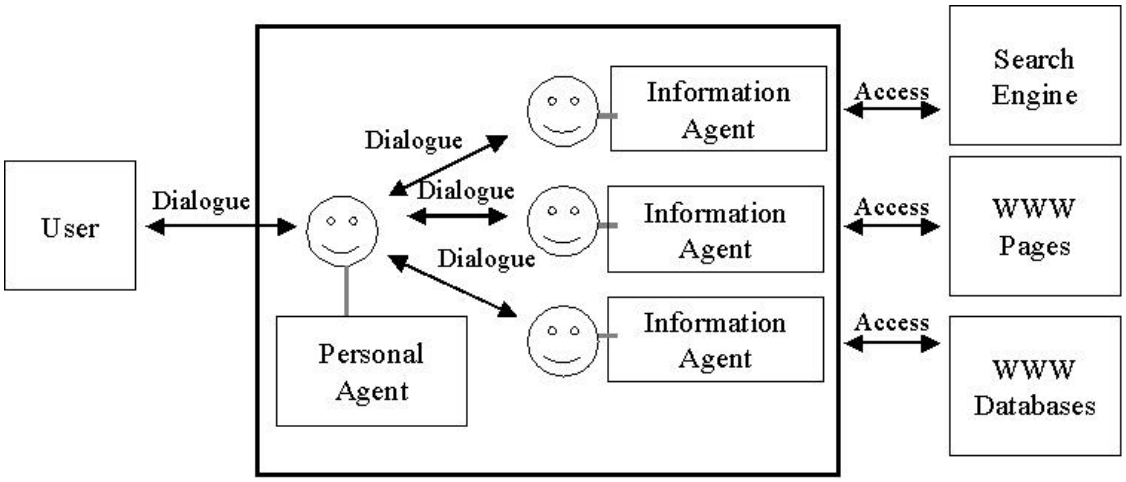

Figure 1: System Architecture.

In our system, to deal with the ambiguity of information sources, we utilize domain specific information agents. An information agent provides information concerning a particular domain such as recipes, restaurants, retailers, and so on. We expect the information agents cannot only provide noiseless information but also facilitate abilities of extracting more detailed information (ex. extracting ingredients from a recipe page) and integrating information from multiple sources (ex. relating ingredients to retailers that carry them).

To deal with the ambiguity of user's query, we utilize the multi-character interface. In the interface, the process of information retrieval is represented to the user through interactions and dialogues among character agents. When an agent makes a mistake about understanding the user's intention, the user can point out the mistake directly to the agent. We expect, through the interaction between the user and the agents, the agents can naturally learn the user's intention.

In the following, we describe "keyword spice", a technology that agentizes a generic search engine into a domain specific information agent, and more about the multi-character interface.

Keyword Spice[1]

A flaw of search engine is that, when a user submits a simple keyword, it returns a large number of various WWW pages that relate to the keyword. To reduce the number of pages, the user usually adds some more keywords, but how appropriate the additional

[1] We are currently studying keyword spice by using a Japanese search engine, so the keyword spices found discussed in this paper may not work well for non-Japanese search engines.

keywords are seems to depend on the experience of the user. For example, when we want to retrieve recipe pages for pork, it is more effective to submit "pork, ingredient" than "pork, recipe." Moreover, the keyword "ingredient" is effective in not only for "pork" but also for other recipe keywords such as "beef" and "chicken." Using this technique, we can build a domain specific search engine (or information agent) from a generic search engine just by adding some keywords, which we call *keyword spices*.

Here we show how to find domain specific keyword spices as follows.

1. Collect a set of WWW pages, which is denoted to be *S*, by using a generic search engine by submitting domain specific keywords (ex. pork, beef, chicken, and so on.)
2. Classify S into the set of domain (ex. recipe) related pages, which is denoted to be *T*, and the set of unrelated ones, which is denoted to be *F*.
3. Select frequent keywords (ex. top 50) that appear in the set T. The keywords are called *keyword spice candidates*.
4. Calculate the appropriateness ratio *A(c)* of each candidate *c*, which is defined to be *A(c)*=(the number of pages in *T* that include *c*)/(the number of pages in *S* that include *c*).

A candidate with the large *A(c)* can be a keyword spice. Multiple keyword spices can be used in a multiplicative way. For example, we can use a keyword spice (ex. ingredient or direction) to retrieve recipe pages. Moreover, we can add more keyword spices to retrieve more specific recipe pages such as Japanese dishes (kelp or mirin can be used as a more specific keyword spice), Western dishes (cream, wine, or butter), or Chinese dishes (sesame oil, ginger)

From a viewpoint of information retrieval, keyword spice is a technique to improve the precision of search engine, sacrificing the recall. To compensate the recall, we can submit an OR combination of multiple spices (ex. cream OR wine OR butter). Hence, if we want to retrieve Western recipe pages for pork, a query can be "pork AND (ingredient OR direction) AND (cream OR wine OR butter)."

Multi-character Interface

In our system, the process of information retrieval is represented as dialogues among the user and the agents as shown in Figure 2.

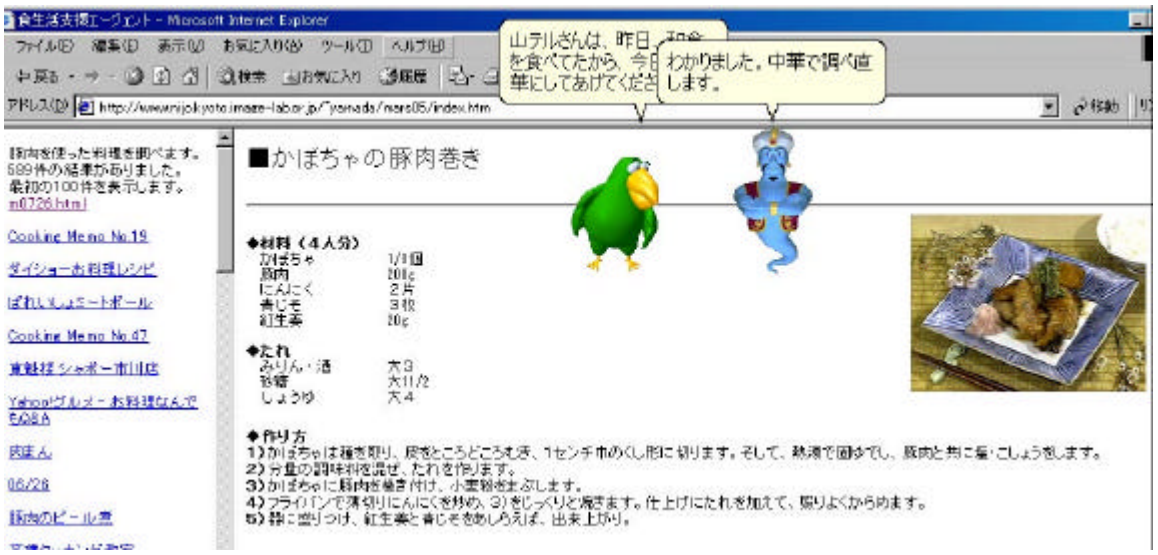

Figure 2: Multiple Character Interface. The servant initially recommends a Japanese dish, but the parrot says "The user cooked a Japanese dish yesterday, so please recommend a Chinese dish today." Then, the servant says "OK. I will recommend a Chinese again."

Character interface is expected to be a more gentle or natural interface to humans than keyboards or GUI because a user can interact with a computer as if he/she communicates with people through dialogue. Moreover, from a business perspective such as e-commerce, characters add more value. For example, Pokemon, Japanese characters born in a computer game, produces a number of related merchandize goods and the amount of annual sales is reported to reach 400 billion yen in Japan and 200 billion yen overseas. Except Pokemon, we have a number of well known characters in Japan such as Hello Kitty, Doraemon, and Momo to name a few. Microsoft also makes a great effort to develop MS-Agent for Windows and, on the Internet, life-like agents developed by Extempo (http://www.extempo.com), Haptek (http://www.haptek.com), Virtual Personalities (http://www.vperson.com), Artificial Life (http://www.artificial-life.com), and so on are now commercially available.

In a conventional character interface, a single character interacts with its user, but we are developing a multi-character interface where multiple characters interact with each other and the user.

An advantage of multi-character interface is that it can contribute to a multi-agent system that tries to learn user's intention or preference. A multi-agent system consists of intelligent agents, each of which has some parameters to be set according to the user's intention or preference to show some intelligent and adaptive behavior, but the problem

is how to learn the intention or preference in a distributed manner. Our multi-character interface can be a solution to this problem. Let us discuss with the following scenario.

(1) User->Personal Agent: "I would like to have a recipe for pork."
(2) Personal Agent->User: "OK. I will call a recipe agent."
(3) Personal Agent->Recipe Agent: "Yesterday you recommended a Japanese recipe, so today please recommend a Chinese recipe for pork."
(4) User->Personal Agent: "I would like to have a Japanese recipe."
(5) Personal Agent->Recipe Agent: "Please recommend a Japanese recipe for pork."
(6) Recipe Agent->User: "How about this?"
(7) User->Recipe Agent: "It looks too much calories. Please recommend another one in low calories."

Here we assume that the personal agent manages a query log and can advise the recipe agent to recommend a type of dishes considering the log and that the recipe agent can recommend recipes considering the calorie. The problem here is that both agents do not know the preference (the type of dish or the amount of calorie) exactly at the initial stage. In the multi-character interface, as the process of recipe recommendation is visually presented through dialogues among agents, so the user can easily notice the mistake of an agent and can point out the mistake directly to the agent as shown in (4) and (7) in the above scenario.

Conclusions

In this paper, we propose a multi-agent based intelligent WWW interfacer. To deal with the ambiguities of information sources and user's query, we developed keyword spice and multi-character interface technologies and show their potentials.

Acknowledgement

This paper reports a part of work done by a joint project of Kyoto University, Osaka City University, SANYO Electric, NTT West, and NTT Comware at LIST (Laboratories of Image Information Science and Technology) subsidized by NEDO (New Energy and Industrial Technology Development Organization). I thank Prof. Toru Ishida for his helpful support as the project leader and Teruhiro Yamada and Takashi Kokubo for their efforts at the development of the system.