XML-based Markup Language for Web Information Integration in 3D Virtual Space

Yasuhiko Kitamura[1], Yatsuho Shibata[2], Keisuke Tokuda[2], Kazuki Kobayashi[2] and Noriko Nagata[1]

[1] *Scool of Science and Technology, Kwansei Gakuin University, 2-1 Gakuen, Sanda, Hyogo 669-1337, JAPAN*
[2] *Graduate Scool of Science and Technology, Kwansei Gakuin University 2-1 Gakuen, Sanda, Hyogo 669-1337, JAPAN*

{ ykitamura, scbc1030, tokuda, kby, nagata}@ksc.kwansei.ac.jp

Abstract

3D virtual space can visually represent the spatial structure to users and an agent can interactively navigate a user in it. The 3D virtual space technology has been applied to city planning, navigation, education, entertainment and so on. The virtual space gets more reality by integrating the Web information. For example, we can change the weather in the virtual space and can open or close the gates according to the corresponding information from the Web. When the virtual space changes depending on the Web information, the agents need to adapt to the change. In this paper, we design an XML-based markup language called AVSML (Agent and Virtual Space Markup Language) to integrate Web information in a 3D virtual space and to specify the behaviors of the agents flexibly. We show how this language can be applied to a campus guide system on VKSC (Virtual Kobe Sanda Campus).

1. Introduction

3D virtual space can visually represent the spatial structure to users and it has been applied to many fields such as city planning, navigation, education, entertainment and so on [5]. In the 3D virtual space, an agent can navigate a user in an interactive manner [6,7]. Various platforms to build a 3D virtual space and languages to control the agents have been proposed [1,2,3,4]. For example, VKSC (Virtual Kobe Sanda Campus) is a 3D virtual space of Kobe Sanda Campus, Kwansei Gakuin University, as shown in Fig. 1 [1]. In VKSC, an agent called Suzie guides a user in the campus upon his/her request.

3D virtual space gets more reality by integrating the real-world information such as the time, weather and the corresponding information of objects in it. Various sensor data can be used to represent the real-world information, and the Web information also can be used though it may be indirect and incomplete real-world information. If the virtual space is tightly connected to the corresponding Web information, it gets more reality [1]. Web information suppliers can inform the latest information to the users through the 3D virtual space and the users experience it in the virtual space.

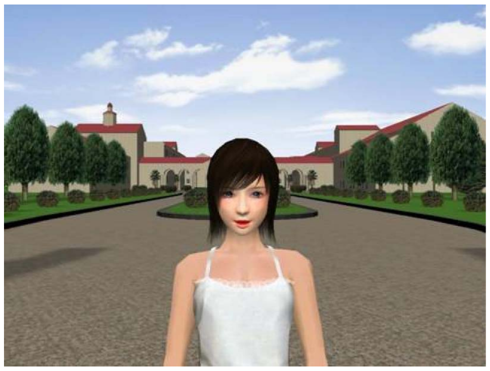
Fig. 1. VKSC.

How to integrate a 3D virtual space and the corresponding Web information is an interesting research issue. It is not good to hard-code the integration process but the process should be open in order that anybody can be involved in the process. We are developing an XML-based markup language called AVSML (Agent and Virtual Space Markup Language) to integrate Web information that is distributed in a number of Web sites into a 3D virtual space.

When a 3D virtual space is tightly connected to the Web information, it changes depending on the update. This means that an agent that inhabits in the space should be adaptive to the change. We are developing a guide agent that autonomously adapts to changes of the 3D virtual space.

In section 2, we discuss Web information integration in 3D virtual space. In section 3, we propose an XML-based markup language to integrate a 3D virtual space and the corresponding Web information. In section 4, we discuss an autonomous agent which guides its user in a changeable virtual space. In section 5, we show how the language and the agent can be used in a virtual campus called VKSC. Finally, we conclude this paper in Section 6.

2. Web information integration in 3D virtual space

We assume that 3D virtual space is composed of objects and agents. Objects are passive entities like buildings, gates and the background. Agents are active entities that can move in the virtual space and interact with the user through chatting.

In our work, the Web information related to the objects in the virtual space can be integrated as shown in Fig. 2. Some Web information can be visually represented in the virtual space. For example, a gate of building in the virtual space can be open or closed according to the Web information about the building and the background changes depending on the weather information from the Web. The other information may not be represented visually, but it can be presented by an agent. For example, an agent can explain that the School of Science and Technology is located in the building.

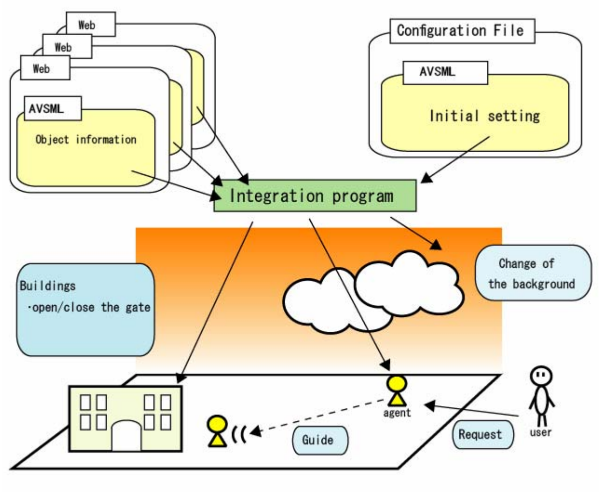

Fig. 2. Integrating Web information in 3D virtual space.

As shown in Fig. 2, the integration program collects the related Web information and reflects it in the 3D virtual space. If a user gives a request to an agent, the agent guides him/her in the virtual space.

To integrate the Web information in a virtual space, we need a language to describe that. It is not good to hard-code the process by using a programming language such as Java, because it is not easy to maintain the relation between the virtual space and the Web information. We are so developing an XML-based markup language called AVSML. XML is easy to be understood by human and easy to be attached to Web pages.

Agents move around in the 3D virtual space and guide a user. If the virtual space changes according to the Web information, the agents need to adapt to the change. We can describe agent behavior as rules in AVSML to cope with this problem. For example, if a gate is closed on the way to a destination, the agent can choose an alternative route that is written in the rules.

3. XML-based Markup Language to Integrate Web Information in 3D Virtual Space

The state of an object is related to the corresponding Web information. For example, the background of virtual space changes depending on the weather information form the Web, and a gate of building is open or closed depending on the opening hours which can be obtained from a Web site of the building.

We use <set> tag to relate the state of an object to the corresponding Web information as below. Objects are identified by objecteID and major attributes are summarized in Table 1.

```
<set object=”objectID”
attribute1=”value1”
attribute2=”value2”…/>
```

Table 1. Major attributes of <set> element.

Attribute	Explanation
State	It specifies the state of the object. (ex. open/closed)
Visible	It specifies whether the object is visible or not. (ex. on/off)
Location	It specifies the location of the object. (ex. entrance)

For example, <set> tags are used as below.

```
<set object=”Main Gate of School of Science
and Technology” state=”closed”/>
```
// The main gate of the School of Science and Technology is closed.

```
<set object=”Notice board”
visible=”on”
   place=”Entrance of School of Science and
Technology”/>
```
// A Notice board appears at the entrance.

We have special objects such as the time and the weather. The appearance and the darkness of the sky in the virtual space change according to the state of the special objects.

```
<set object="time" state="12"/>
// The time is set to be 12.

<set object="weather" state="fine"/>
// The weather is set to be fine.
```

We can set the time by using a NTP server and set the weather by using a weather information Web site. We use a wrapper program to extract the weather information related to the virtual space from a RSS site or a general Web site.

```
<set object="time" state="ntp_server(URL)"/>
// The time synchronizes to the NTP server.

<set object="weather" state="wrapper(URL)"/>
// The weather information is acquired through a Web site specified
by URL.
```

The state of an object may change depending on the time or the other conditions, so we introduce <if> and <then> tags. In <if> element, we can insert a logical expression using <and> or <or> tag and their combinations. For example, if the main gate of School of Science and Technology building is open only from 8AM to 8PM in weekdays, we specify that as below. <test> tag is used to check if the attribute value is true or not.

```
<if>
  <or>
    <test object="day" state="Saturday"/>
    <test object="day" state="Sunday"/>
  </or>
  <then>
   <set object="Main Gate of School of Science
and Technology" state="close">
  </then>
</if>
<if>
  <and>
    <or>
     <test object="day" state="Monday"/>
     <test object="day" state="Tuesday"/>
     <test object="day" state="Wednesday"/>
     <test object="day" state="Thursday"/>
     <test object="day" state="Friday"/>
    </or>
    <test object="time"
state="greater_than(8)"/>
    <test object="time"
state="less_than(20)"/>
  </and>
 <then>
  <set object=" Main Gate of School of
Science and Technology" state="open">
  </then>
</if>
 <and>
   <test object="time" state="less_than(7)"/>
    <test object="time"
state="greater_than(21)"/>
    <and>
    <then>
      <set object=" Main Gate of School of
Science and Technology" state="closed">
    </then>
```

Some part of Web information such as the history of a building cannot be represented visually and it is suitable to be represented by an agent. We introduce <info> tag for an agent to explain the Web information.

```
<info object="Building of School of Science
and Technology">
  This building is the School of Science and
Technology. It is built in 2001.
  </info>
```

4. Guide agent

Because the virtual space surrounding agents may change according to the Web information, the agents are needed to adapt to the change. We therefore introduce rules to describe flexible actions of agents. Upon a request from the user, the agent creates a plan to achieve the request by using the rules. Below is an example of rule.

```
<rule>
  <implies>
   <head>
    <task type="guide" location="School of
Science and Technology"/>
   </head>
   <body>
    <ptask type="move" location="School of
Science and Technology"/>
    <ptask type="speak" object="School of
Science and Technology"/>
   </body>
  </implies>
</rule>
```

A <rule> element has one or more <implies> elements, and an <implies> specifies one rule that consists of <head> and <body>. In this example, a task <task name="guide" location="School of Science and Technology"/> is decomposed into two subtasks <ptask name="move" location=" School of Science and Technology"/> and <ptask name="speak" object=" School of Science and Technology"/>. <ptask> means a primitive task that cannot be decomposed further but can be executed by an agent.

To cope with the change of virtual space, a rule can contain <if> and <then> structure. If the main gate of the School of Science and Technology is open, the agent can use the main gate to enter the building. Otherwise, it uses the sub gate of the building.

```
<rule>
  <implies>
    <head>
```

```
        <task type=”enter” location=”School of
Science and Technology”/>
      </head>
      <body>
       <if>
         <test object=”Main Gate of School of
Science and Technology” state=”open”>
         <then>
           <ptask type=”move” location=”Main Gate
of School of Science and Technology”/>
         </then>
        </if>
       <if>
        <test object=”Main Gate of School of
Science and Technology” state=”open”>
        <then>
         <ptask type=”move” location=”Sub Gate of
School of Science and Technology”/>
        </then>
       </if>
      </body>
    </implies>
   </rule>
```

5. VKSC Guide System

In this section, we show how we can integrate Web information in a 3D virtual space by using an example of VKSC (Virtual Kobe Sanda Campus) guide system. VKSC is a virtual campus created by computer graphics and an agent called Suzie guides the user around the campus.

The integration program initially reads the configuration file like below. The file specifies the URLs for special objects and those for the Web information to be integrated using <seealso> tags. In this example, three URLs; http://www.kwansei.ac.jp/ (Kwansei Gakuin University), http://sci-tech.ksc.kwansei.ac.jp/ (School of Science and Technology) and http://ist.ksc.kwansei.ac.jp/ (Department of Informatics) should be integrated. Rule description also may be included in the configuration file.

```
   <?xml version=”1.0”?>
   <set object=”time”
state=”ntp_server(time.windows.com)”/>
   <set object=”weather”
state=”wrapper(http://weather.livedoor.com/forec
ast/rss/28/82.xml)”/>
   <seealso url=”http://www.kwansei.ac.jp/” />
   <seealso url=”http://sci-
tech.ksc.kwansei.ac.jp/” />
   <seealso url=”http://ist.ksc.kwansei.ac.jp/”
/>
```

Initially, the user gives a request such as
<task type=”guide” location=”Department of Informatics”/>
to Suzie. Suzie creates a plan to guide the user by using the rule description. If the main gate of the School of Science and Technology is open, then the plan becomes

```
   <ptask type=”speak” object=”Kwansei Gakuin
University”/>
   <ptask type=”move” location=”School of Science
and Technology”/>
   <ptask type=”move” location=”Main Gate of
School of Science and Technology”/>
   <ptask type=”speak” object=”School of Science
and Technology”/>
   <ptask type=”move” location=”Department of
Informatics”/>
   <ptask type=”speak” object=”Department of
Informatics”/>.
```

Suzie starts to introduce Kwansei Gakuin University to the user referring to the <info> information at http://www.kwansei.ac.jp/. She then moves to the front of School of Science and Technology and introduce the school referring to the <info> information at http://sci-tech.ksc.kwansei.ac.jp. Finally, she enters the building through the main gate, moves to the Department of Informatics, and introduces the department referring to the <info> information at http://ist.ksc.kwansei.ac.jp. If the main gate of School of Science and Technology is closed, Suzie enters the building through a sub gate.

6. Conclusion

In this paper, we proposed an XML-based markup language AVSML (Agent and Virtual Space Markup Language) to integrate Web information in 3D virtual space. A virtual space changes depending on the corresponding Web information and the virtual space gets more reality. On the other hand, agents in the virtual space need to adapt to the change. We adopt rules that agents’ actions can be described flexibly, so that the agents can cope with this problem.

By introducing an XML-based language, it is easy to write and understand how Web information is integrated in a virtual space and how agents behave in it. Another advantage is that the description can be attached to Web pages written in HTML, so it can be easily updated.

References

[1] Y. Kitamura, et al. Toward Web Information Integration on 3D Virtual Space. ICEC2005, LNCS 3711, Springer, 445-455, 2005.
[2] M.Nischt et al. MPML3D: A Reactive Framework for the MPML, IVA2006, LNAI 4133, Springer, 218-229, 2006.
[3] T. Ishida. Q: A Scenario Description Language for Interactive Agents. IEEE Computer, 35(11): 42-47, 2002.
[4] T. Ishida. Digital City Kyoto: Social Information Infrastructure for Everyday Life, CACM, 45(7):76-81, 2002.
[5] Lars Qvortrup (ed.) Virtual Space: Spatiality in Virtual Inhabited 3D Worlds, Springer, 2002.
[6] H. Prendinger and M. Ishizuka (eds.) Life-Like Characters: Tools, Affective Functions, and Applications, Springer, 2004.
[7] J. Cassell, et al. (eds.) Embodied Conversational Agents, MIT Press, 2000.