

Co-presence, collaboration, and control in environmental studies

A Second Life-based approach

Songpol Attasiriluk · Arturo Nakasone ·
Wisut Hantanong · Rui Prada ·
Pizzanu Kanongchaiyos · Helmut Prendinger

Received: 5 February 2009 / Accepted: 17 June 2009 / Published online: 5 August 2009
© Springer-Verlag London Limited 2009

Abstract In this paper, we describe a framework for synchronous collaborative visualization and remote control in the agricultural domain. The framework builds on “Second Life” (SL), a popular networked online 3D virtual world, where users are represented as avatars (graphical self-representations). Co-presence in SL takes the form of instant (real-time) two-way interaction among two or more avatars. The aim of our work is to facilitate co-presence for sharing knowledge and exchanging wisdom about environmental practices. In order to establish a realistic simulated context for communication in SL, virtual counterparts

of real devices are created in the virtual world. Specifically, we aim to represent field servers that sense and monitor fields such as rice paddies and vineyards. The Twin-World Mediator (TWM) is developed in order to replicate the behavior of real devices in virtual counterparts, and to facilitate seamless communication between real and virtual world. The TWM is an easy-to-use, extensible, and flexible communication framework. A small study demonstrated how the TWM can support collaboration and experience sharing in the agricultural domain.

Keywords Presence in shared virtual environments and online communities · Avatars · Presence applications (communications and collaboration; teleoperation)

S. Attasiriluk, A. Nakasone, W. Hantanong contributed equally to the content of the paper.

S. Attasiriluk and W. Hantanong were awarded a Memorandum of Understanding (MOU) Internship grant from the National Institute of Informatics, Tokyo.

S. Attasiriluk · A. Nakasone · W. Hantanong ·
H. Prendinger (✉)
National Institute of Informatics, 2-1-2 Hitotsubashi,
Chiyoda-ku, Tokyo 101-8430, Japan
e-mail: helmut@nii.ac.jp

A. Nakasone
e-mail: arturonakasone@nii.ac.jp

W. Hantanong
e-mail: wizzup@wizzup.com

S. Attasiriluk · W. Hantanong · P. Kanongchaiyos
Department of Computer Engineering,
Chulalongkorn University, Bangkok, Thailand
e-mail: pizzanu@cp.eng.chula.ac.th

R. Prada
IST-UTL, INESC-ID, Av. Prof. Cavaco Silva,
Taguspark, 2744-016 Porto Salvo, Portugal
e-mail: rui.prada@gaips.inesc-id.pt

1 Introduction

In everyday life, humans share experiences by attending to or interacting with objects within the same reference frame. For example, a group of fans cheer their sports team in the stadium, company workers discuss their new product design shown on a display, and children play a video game on the same console. Those kinds of activities require a two-way interaction among various people on the same object. Typically, people have to “be there” at the same physical location to ensure that they are sharing the same experience. However, the development of communication networks such as the telephone and Internet has been successful in introducing alternative ways of experience sharing for geographically separated people. For instance, applications such as online-gaming and teleconferencing implement the concept of “virtual co-presence” where communication partners are present as virtual simulations (Zhao 2003).

The concept of “presence” is extensively described in Lombard and Ditton (1997). The authors show that presence can refer (1) to the degree that a communication medium is perceived as sociable or personal, (2) to the extent that real-world objects and events are represented accurately, (3) to the idea of ‘transportation’: “being there” or “being together” (in the simulated environment), (4) to the concept of perceptual and psychological immersion, e.g. via head-mounted display, (5) to social interfaces with embodied social actors, and even (6) to anthropomorphic cues of the medium itself. In our approach, the notion of presence is anchored to the user’s avatar, i.e. his or her graphical self-representation in the virtual world.

According to the taxonomy suggested by Zhao (2003), our platform can be conceived as an application that supports hypervirtual telepresence, where persons are presented in electronic proximity through digital simulation. In this form of co-presence, humans can entertain immediate (synchronous) interactions and receive feedback through the interlocutor’s avatar gesture and behavior. An important aspect of co-presence in Second Life (SL) is that multiple users (as avatars) can share the same situation and experience.

With the rapid growth of the web, social networking web sites are becoming the prime collaboration and advice-giving work spaces (Weaver and Morrison 2008). Millions of people spend their time online to offer or receive help from their online contacts, which can be seen as a form of asynchronous collaborative work. A large number of Web2.0 sites were created to support collaborative work between remote people, e.g. in medicine (Eysenbach 2008). Despite the improvement in connecting experts with help seekers, the communication in current social network sites still lacks the immediate nature of face-to-face communication or presence.

Since the new generation of personal computers can bring the performance of realistic 3D interfaces to common users at a reasonable price, 3D networked virtual worlds have been suggested as affordable collaborative platforms (Bray and Konsynski 2007; Bainbridge 2007). In particular, SL, a 3D multi-user online virtual world, became very popular in 2007, with over 15 million registered users (as of September 2008) and about 40,000 users logged in at any time. SL attracts significant interest from various communities including anthropology (Boellstorff 2008), medical and health education (Boulos et al. 2007), and location-based systems (Brandherm et al. 2008; von Kapri et al. 2009).

However, SL is not yet established as a common platform for advanced collaborative work. We speculate that the need to learn the libsecondlife software library (libsecondlife 2007) and the “Linden Scripting Language” (LSL, Weber et al. 2007) is a major hindrance for common

users without dedicated programming skills. Hence, we developed the Twin-World Mediator (TWM), a software framework which supports ready-to-use communication management between the real and virtual world. In particular, we want to facilitate collaboration, knowledge exchange, and control of a sensor–actuator device in environmental studies.

The rest of the article is structured as follows. In Sect. 2, we present a brief review of related work. Section 3 describes two SL-based scenarios of collaboration in agriculture. The scenario involving the control of a field server motivates the TWM communication framework, which will then be described in detail in Sect. 4. In Sect. 5, we describe the results from a pilot study about a knowledge sharing in agriculture. Section 6 discusses technical limitations and Sect. 7 concludes the paper.

2 Related Work

Here, we report on some related work in the categories virtual reality (VR), collaborative virtual environments, and connection between real world and SL.

The research field of VR has proposed interfaces that let users interact immersively with objects (Bryson 1996) and other users (Ogi et al. 1999). Milgram et al. (1994) investigated ‘mixed reality’ displays of the reality–virtuality continuum, such as augmented reality and augmented virtuality interfaces. Similar to our SL-based approach, VR aims to support tele-presence and associated interaction capabilities with things and people. However, while the fidelity of interaction in a virtual world might be lower than, e.g., in a 3D cave, the interaction capability is affordable to any networked computer user.

Work in collaborative virtual environments emphasizes the benefits of a shared 3D view. Leigh and Johnson (1996) developed the concepts of shared viewpoint and telepointer to facilitate cooperation. However, natural face-to-face communication is not supported. The work of Valin et al. (2001) recognizes the importance of persistent virtual environments, but like other VR applications, it depends on special-made equipment.

Finally, there is some preliminary work on connecting real-world devices to SL. In Lifton et al. (2007), data from a specialized power-plug-based sensor network is fed into the virtual world by means of a SL script-based implementation of a XML-RPC protocol. However, the data are used for visualization only and there is no support for interaction with it. Fundinger (2007) reports about a real-world control panel that can both control and be controlled by objects in SL, i.e. changes to the knobs or push buttons in the real world are translated to virtual buttons, which control the LEDs on the real-world control panel. The

control panel can be seen as a specific implementation, whereas the TWM provides a generic solution to two-way communication. The first steps towards a generic solution are described in Brandherm et al. (2008).

3 Motivating example: participatory agriculture

Participatory agriculture is an example for co-presence (more precisely, hypervirtual telecopresence) in an online virtual world with the purpose of sharing information about the real world (von Kapri et al. 2009). The concept of “participation” is not new in agro-ecosystem analysis. In pest analysis, e.g., it is practiced using a pen and paper approach, in which groups of farmers are taken to a crop field and asked to draw the type of plant they see on a sheet of paper, along with the insects that they can identify as being beneficial or harmful to the crop (Mangan 1997). Furthermore, the infrastructure for environmental data acquisition is available through the implementation of networked field servers (distributed sensing devices composed of a web camera, multi-sensors, a wireless LAN module and a high intensity LED lighting to be used for environmental measurement, plant/animal monitoring, and farm fields/facilities observation), see e.g. Du et al. (2008).

Inspired by real-world objects (a rice paddy, field servers), we have created virtual meeting spaces in SL where information relevant to environmental studies is displayed and can thus be shared concurrently by all visitors, e.g. experts (senior farmers) and novices (junior farmers). Furthermore, we have created virtual (counterpart) devices, with which SL avatars can interact in the same way as people would operate the real device in the real world. Using the TWM described in this paper, we could easily create the behavior of all virtual devices, which either display information or allow avatars to interact with them (see Sect. 4). In our demonstrators, we use information from the National Agricultural Research

Center (NARC) field server and Shinshu University field server.

The field server of the NARC in Japan provides real-time video images and agricultural data, including air temperature, humidity, soil temperature, soil moisture, CO₂ concentration, leaf wetness, etc. We have created a virtual rice paddy with a virtual version of the NARC field server (see Fig. 1a). This environment supports working meetings of agricultural experts, owners of agriculture fields, and layman in order to conveniently discuss the current status of the field across physical borders and make decisions about its treatment.

The field server of Shinshu University provides real-time image, temperature, and humidity. We created a virtual version of this field server, which can be used to control the real camera (see Fig. 1b). For example, if a visitor pans the virtual camera, the new camera image is retrieved from the real camera and refreshed in the display screen. As in the case of the previous scenario, interested persons can meet virtually and discuss based on the shared situation.

4 The Twin-World Mediator

The TWM framework is a robust platform for interconnection between real-world devices and their virtual counterparts. Even though the basic functionality of the framework is the transmission of data from the real world to the virtual world and vice versa, we took special consideration in its design, providing common users a tool, i.e.:

- *Easy to use.* Non-programmers can create virtual devices using a simple description file written in an XML-based language.
- *Extensible.* Both the description of real devices and the communication channels are independent of the



(a) Virtual NARC Field Server.



(b) Virtual Shinshu Univ. Field Server.

Fig. 1 Two scenarios for environmental studies. **a** Visual representations of temperature and humidity visualizations (10 each) based on data of Yamani Orchard, Ichikawa (data captured on 26 January 2009; 15:00), based on Ichikawa (2009). **b** Number of panels for

temperature and humidity (data captured on 26 January 2009; 15:00), based on Kobayashi (2009). The *right-most avatar* operates (‘rides’) the controller of the field server

application program, which makes the addition or deletion of real devices and their virtual representations very easy to perform.

- *Flexible*. Users can select SL objects of their choice to customize the way they visualize the data that come from the real devices.
- *Secure*. TWM provides a two-bot data communication paradigm, which helps to prevent the misuse of the framework to transmit incorrect or malicious data in the form of spam.

4.1 TWM architecture components

The TWM framework is composed of two independent executables: the *TWM Client application* and the *TWM Server application*, which can be run on separate computers. The TWM Client application is for device owners who want to connect their devices to the virtual world, whereas the TWM Server application is for SL island owners who provide the space for the creation of their paired virtual devices (see Fig. 2).

The TWM Server application is in charge of instantiating and managing the virtual devices in SL. In order to perform this task, it logs in a computer-controlled avatar (i.e. the Server Bot), through which it issues commands framed as instant messages to create and modify the objects that would represent the devices. When this Server Bot receives device manipulation requests from a TWM Client application, it utilizes the Virtual Device Storage database to store specific information that helps the program determine which virtual devices correspond to this particular client. In addition, when any modification to a virtual

device is perceived, the modification is transmitted to the Server Bot as an instant message and retransmitted back to the corresponding TWM Client application.

The TWM Client application is in charge of connecting the real devices to the virtual world. Similarly, it logs in a computer-controlled avatar (i.e. the Client Bot), through which it transmits device manipulation requests to the TWM Server application. When the application starts, it reads the information regarding real devices' connection methods and data formats from a configuration file called the Device Description File (DDF). Once the Client Bot is logged, it requests a connection with a predefined Server Bot and, when the connection is established, the TWM Client application starts issuing the appropriate commands to (1) create the virtual devices that correspond to each real device defined in the DDF and (2) update the virtual devices' values based on the real ones. When any virtual device update notification comes from the Server Bot, it is perceived by the Client Bot and retransmitted to its correspondent real device by the application.

The communication between the Server Bot with the Client Bot is implemented as an XML-based instant message passing protocol known as the Communication Description Language (CDL), which is described in detail in Sect. 4.5.

Both the TWM Client and the TWM Server applications were implemented in C sharp, making use of a third party connection library called OpenMetaverse (2009), which allows external programs to manipulate avatars and objects in SL.

4.2 Sensor/actuator definitions

The TWM has two categories of devices:

1. *Sensors*. These are devices that sense real-world data and display them on their virtual world counterparts. An example of a sensor can be an electronic device connected to a computer or any kind of device that provides a way to retrieve data from it.
2. *Actuators*. These are devices in which avatars can interact with, such that the result of interaction (in SL) drives their real-world counterparts to perform a task. Interactions in SL include touching, turning, driving, etc. An example of an actuator is a 3D visual component in SL.

Figure 3 shows four examples of virtual devices we have created for our testing scenarios. Figure 3a and b represents sensors and Fig. 3c and d represents actuators. Figure 3a depicts a virtual chart to display multidimensional numeric values. This chart displays ten prisms, with the height of each prism representing a value between 0 and 100. Figure 3b is a seven-segment display of

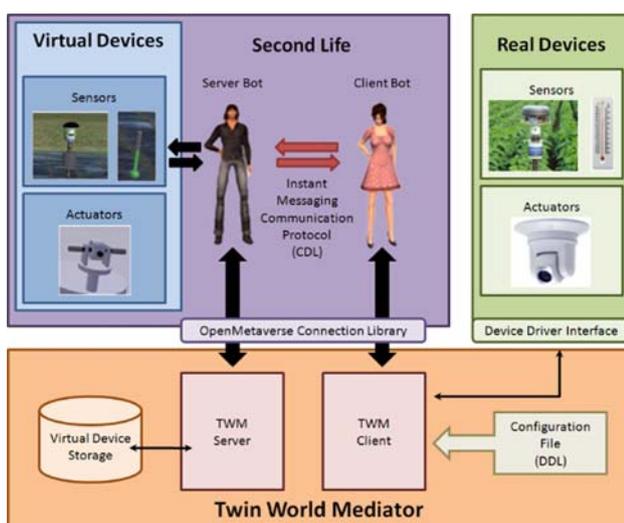


Fig. 2 TWM architecture design: the client component manages the connection to the real devices, whereas the server component manages their virtual counterparts

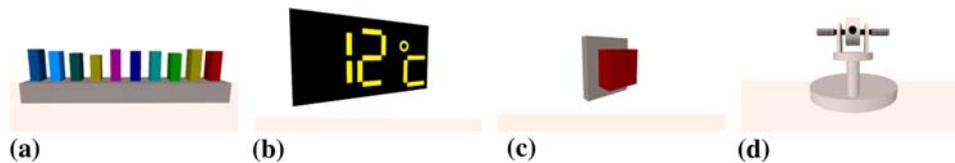


Fig. 3 Examples of virtual devices. **a**, **b** are of type sensor, and **c**, **d** are of type actuator. **a** Chart display for data array, **b** temperature display for thermometer, **c** ON/OFF switch controller, and **d** camera controller

temperature data in Celsius units. Figure 3c represents a push switch that returns a boolean value (i.e. pushed or not pushed) to any actuator device, and finally, Fig. 3d is a camera controller that returns pan values when an avatar ‘rides’ on it and the user presses the arrow buttons on the keyboard.

The TWM framework was not designed for the requirements of only specific devices. On the contrary, it was designed to be scalable and open for the inclusion of new virtual and real devices. All communication to devices, both real and virtual, is performed through device interfaces. For real devices, the device interface is called the “device driver”. The most common way to read data from or write data to a real device is through the use of a network protocol (e.g. HTTP, FTP), a file, or an I/O port (e.g. serial, parallel). For virtual devices, the device interface takes the form of chat messages over certain chat channels. Virtual devices require embedded LSL scripts in order to be able to send data, receive data and update their appearance (color, size, etc.). The TWM framework can interact with any real device that can provide any of the interfaces described above and any virtual device that implements a messaging system which the Server Bot can understand.

Since most real devices can be classified into sensors and actuators, the TWM was designed to provide functional connectivity on the sensor/actuator paradigm, making it very easy for any equipment to work with TWM. Device owners just need to write a detailed parametric description of their device. TWM allows definitions specified in common data types found in every programming language (e.g. integer, floating point number, character string, and arrays).

4.3 TWM connection scenario

When users want to establish a connection between their real devices and SL, the procedure defined below has to be followed:

- The user who wants to connect his/her real devices (i.e. the device owner) needs to specify beforehand the description of each device, and how to connect to it from the client application. This is performed through the creation of an XML-based configuration file called

the DDF. Also, the user needs to provide information regarding the Client Bot which will stream the real data into SL and the Server Bot which will receive the data and update the virtual devices.

- Accordingly, the user in charge of manipulating the Server Bot needs to specify the list of Client Bots that will be allowed to communicate with his/her Server Bot in a XML-based configuration file.
- When the Client Bot logs into SL, it will start requesting the creation of a virtual counterpart for every real device listed in the client’s configuration file. If the device is not present in the virtual world, it will be instantiated from the Server Bot’s inventory to the position specified by the client.
- After the Client Bot receives an acknowledgement that the virtual device connection has been established, the communication with the Server Bot will begin, with the direction of the communication depending on the type of device instantiated. If the device is defined as a sensor, it waits for data from the Sensor Bot. If the device is defined as an actuator, it will wait for the user to manipulate the virtual device.
- To update a sensor device, the Client Bot periodically reads the data from the real device and sends it to the Server Bot, which updates the virtual device.
- When an avatar, with the proper permissions given by the island owner, activates (either by pushing, driving, etc.) the virtual actuator, this condition will be detected by the Server Bot that will transmit the information to the Client Bot, which will pass the data to the real device.
- The two-way communication between Client and Server Bot continues until one of them logs out of the virtual world.

4.4 Device Description File

The TWM framework provides a way to easily configure real devices for client application users through the specification of an XML-based configuration file called “DDF”. This file is written using a proprietary language created by us known as the Device Description Language (DDL) (GlobalLab 2008–2009). Most real devices can easily be specified using DDL, covering the majority of demands,

and users can also add their own specifications with little effort. Our language is conceptually similar to the sensor definition language called SensorML (2009), but we extended this language to include actuator definitions.

A typical DDL file has three major structural components: the *login* section, the *sensors* section, and the *actuators* section. A simple configuration file which contains only one sensor and one actuator is shown in Fig. 4.

In the login section, indicated by the *login* and *serverbot* tags, the user must specify the login information for the Client Bot (its first name, last name, and password) as well as for the Server Bot to which the client application will connect to. In this case, the Server Bot is identified by its assigned UUID (Universal Unique Identifier), which must be retrieved from SL.

The sensors section, indicated by the *sensors* tag, describes the list of physical sensors that will be displayed as virtual objects in SL. Each sensor is defined by a unique id, a name, a visual representation (the type of predefined object used for the sensor in SL), its location in the virtual world, and the type of data value it generates, along with the source of such data. In the sample configuration file, the sensor defined as “1” is a device that provides single integer values that would be displayed as a thermometer in the virtual world.

```

1 <clientbot version="1.0"
2   xmlns="http://www.prendingerlab.net/sensorbot/client">
3   <login>
4     <firstname>TWM</firstname>
5     <lastname>Client</lastname>
6     <password>...</password>
7     <location island="NIIIsland" x="160" y="236" z="34" />
8   </login>
9   <serverbot>
10    <uuid>20bf5857-e65b-4cbe-8d9c-...</uuid>
11  </serverbot>
12  <sensors>
13    <sensor id="1" mode="push" name="Local thermometer"
14      visual="THERMOMETER">
15      <placement>
16        <position x="162" y="235" z="27" />
17        <rotation x="0" y="0" z="90" />
18      </placement>
19      <value type="int" dimension="1" refresh_int="10s">
20        <source protocol="HTTP" uri="http://p_to/sens1"/>
21      </value>
22    </sensor>
23  </sensors>
24  <actuators>
25    <actuator id="2" mode="push" name="Local switch"
26      visual="ONOFF">
27      <placement>
28        <position x="162" y="236" z="27" />
29        <rotation x="0" y="0" z="0" />
30      </placement>
31      <value type="bool" dimension="1" initial="false">
32        <target protocol="FILE" uri="p_to/_onoff.txt"/>
33      </value>
34    </actuator>
35  </actuators>
36 </clientbot>

```

Fig. 4 Simple DDL configuration file for the TWM Client application

Lastly, the actuators section, indicated by the *actuators* tag, describes the list of physical actuators that will be represented as manipulable virtual objects in SL. Each actuator is defined by a unique id, a name, a visual representation, its location in the virtual world, and the type of data value it accepts, along with the target of such data. In the sample configuration file, the actuator defined as “2” is a device that can be set through a boolean value and that would be displayed as an ON/OFF switch.

4.5 Client server communication protocol

The Client Bot and the Server Bot communicate with each other by sending XML-based messages using the SL instant messaging channel. The specification of this message is written in what we call the “CDL”. This proprietary language (also developed by us) helps to describe the target, purpose of the message, and details of the operation to be performed in the virtual device (GlobalLab 2008–2009). CDL was designed to be open to developers who want to build their own Client or Server Bot.

As an example, Fig. 5 shows a sample CDL message that is sent from a client application to create a virtual sensor. In contrast, Fig. 6 shows a sample CDL message that is sent from a server application to update a real actuator device.

A typical CDL message, identified by the *conversation* tag, is composed of a timestamp, which records the time in which the message was generated, and an instruction or instructions (identified by the *message* tag) which will be executed either in the virtual device or in the real device, depending on the direction of the message.

Any instruction in CDL has three main attributes: the type of device to which the instruction is targeted (sensor or actuator), the type of instruction to execute, and its parameters. Instructions for virtual device operation

```

1 <conversation version="1.0"
2   xmlns="http://www.prendingerlab.net/sensorbot/msg">
3   <timestamp>1/25/2009 2:21:57 PM</timestamp>
4   <message>
5     <about>SENSOR</about>
6     <for>CREATE</for>
7     <detail>
8       <id>1</id>
9       <name>local Thermometer</name>
10      <type>THERMOMETER</type>
11      <placement>
12        <position x="162" y="235" z="27" />
13        <rotation x="0" y="0" z="90" />
14      </placement>
15      <value>0</value>
16    </detail>
17  </message>
18 </conversation>

```

Fig. 5 Client to server: CDL client message for creating device

```

1 <conversation version="1.0"
2   xmlns="http://www.prendingerlab.net/sensorbot/msg">
3   <timestamp>1/25/2009 2:21:57 PM</timestamp>
4   <message>
5     <about>ACTUATOR</about>
6     <for>UPDATE</for>
7     <detail>
8       <id>2</id>
9       <value>>true</value>
10    </detail>
11  </message>
12 </conversation>

```

Fig. 6 Server to client: CDL server message for updating client's actuator

between the server and the client applications can be divided into four types:

1. *Virtual device creation.* The client application requests to establish a connection with a specific virtual device on SL.
2. *Virtual device update.* For sensors, the client application sends updated values to the server application, including new position and new rotation values. For actuators, the server application sends updated data to the client application.
3. *Virtual device removal.* The client application requests to remove a virtual device from the virtual world.
4. *Operation acknowledgement.* The sensor application sends an acknowledgement back to the client application that the creation, update or removal operation was either successful or not.

The parameters of the instruction, identified by the `detail` tag, can vary depending on the type of instruction. For instance, a create operation requires the specification of an id, a name, a visual representation, a location and a value; on the other hand, an update operation just requires the id of the device and a value.

5 Pilot study

We prepared a simple scenario using our virtual devices in SL, and tested the response of users to the in-world experience. In the scenario, users, in pairs, were asked to discuss and make a decision regarding an agricultural problem. They presented some data, provided by virtual devices, to facilitate their discussion, and four different options to choose from as a means to save the field. The virtual devices provided measures of temperature and humidity in the field, with a history of the past 7 days, and an image from the field that could be explored by panning the virtual camera.

Subjects were shown a situation with a dry field and some humid and hot days. The four choices were (1) to add water, (2) to reduce the sun exposure, (3) to apply pesticide

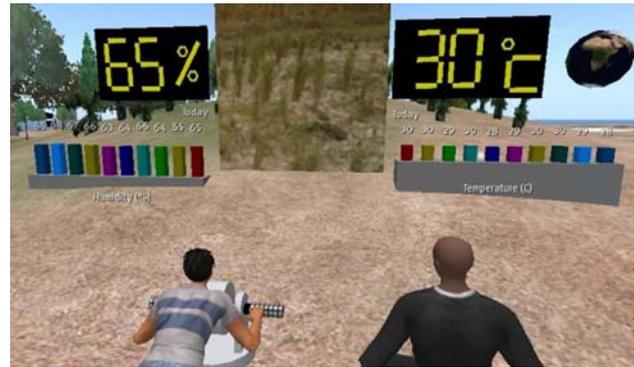


Fig. 7 The left avatar is controlling the camera. Both avatars co-experience humidity (left), a view on the field (middle), and temperature (right). Humidity and temperature are displayed for 10 days

or (4) to do nothing. The situation was controlled in order to be ambiguous and generate some discussion. The actual choice made in the end was not important. Figure 7 shows the experiment setup in SL.

The experiment was conducted with 10 subjects (5 pairs), students at National Institute of Informatics, with ages between 21 and 35 (mean 27.3, $\sigma = 3.83$). All except one subject were male. Most subjects did not have prior experience with SL even though they have some experience with virtual environments in general. When asked about their expertise in agriculture, they confirmed to have some knowledge¹ (e.g. common sense). Subjects were instructed to discuss and come to a decision within 10 min. Time spent in the experiment was in average 6 min 14 s ($\sigma = 1$ min 48 s). Therefore, all groups successfully performed the task within reasonable time.

To assess the experience of users, we applied a co-presence questionnaire developed by Casanueva and Blake (2001), consisting of 6 questions ranked in a 1–7 Likert scale (see Table 1). In addition, we recorded the time spent for the experiment and the chat log.

Results show that a sense of co-presence was successfully achieved. We obtained a mean rank of 5.23 ($\sigma = 0.63$) in the co-presence questionnaire. Furthermore, by carefully analyzing each individual question, we realized that the question concerned with the emergence of group/community was ranking quite low (mean rank 4.20, $\sigma = 2.0$). We speculate that by using pairs of subjects rather than bigger groups, this sense of group emergence was broken. If we remove this question in our measure of co-presence, our results are slightly improved (mean rank 5.44, $\sigma = 0.42$). In fact, the questionnaire's Cronbach's α

¹ This was assessed by asking subjects to rate their expertise in agriculture in a seven-point Likert scale (1 representing “no knowledge at all” and 7 representing “professional knowledge”). The mean value for this was 3.6 ($\sigma = 1.2$).

Table 1 The items of the co-presence questionnaire

-
1. To what extent did you have a sense that the other person of the team was in the same place as you during the course of these events?
I sensed that the other person was in the same place as me...
 2. To what extent did you have a sense that you were in the same place as the other person during the course of the experience?
I sensed that I was in the same place as the other...
 3. To what extent did you have a sense of the emergence of a group/community during the course of these events?
I sensed the emergence of a group...
 4. To what extent did you have a feeling that you were collaborating with real people and not robots?
I had a feeling that I was collaborating with real people...
 5. When you think back about your last experience, do you remember this as more like talking to a computer or communicating with people?
I remember this as communicating with people...
 6. To what extent did you have a sense of being “part of the team”?
I had a sense of being “part of the team”...
-

Items were ranked in a Likert scale of 1–7 (1 meaning “never” and 7 meaning “all the time”)

improved significantly, from 0.65 to 0.79, after removing the question.

Regarding the chat log, we found that the discussion was usually dominated by one person although both conversed. The average number of chat messages was 26.0 ($\sigma = 18.99$). The average number of chat messages of the more talkative person was 17.8 ($\sigma = 11.99$), while for the less talkative it was 8.6 ($\sigma = 7.2$). The discussion was generally centered in the choice between to add water to the field and to cover the field to reduce sun exposure. Both persons, in each pair, presented arguments based on the readings of the virtual sensors, but, usually, only one controlled the camera. This can be explained by the fact that the camera’s controls could only be used by one person at a time. In the end, three of the groups decided to add water to the field while the other two decided to cover the field to reduce sun exposure. Both answers are acceptable given the data that were shown to the users.

Here are some free-text comments from the subjects. One participant stated “In the beginning I was not aware, that another user was on the island as well. But then the collaboration was very interesting, I had the feeling like I talk to him directly face-to-face”, while another said “If graphics could be improved and speech could be added (e.g., TTS), we could have better experience. But it is a great way of experiencing the web based content.”. These comments indicate that the experience was generally good, even though the fidelity of the virtual environment is not high.

6 Discussion

For common users, the TWM framework provides a flexible and easy-to-use approach to display data of real devices in SL. Nevertheless, some technical issues regarding the external connectivity library (libsecondlife) and SL itself limited our options to deliver a more streamed

solution. For example, the information shown in our virtual screen in the rice field corresponds to a texture image applied over the screen itself, whereby a real-time screen would require the display of several different texture images per second. However, Linden Labs (the company running SL) charges a small amount of real money to upload images. Thus, we used another feature offered by Linden Labs called MediaTexture. Here, a user can stream information content into the island via one unique channel. This feature nicely overcomes the economic issue, but constrains TWM into processing just one stream at a time per island.

In addition, working with libsecondlife created some unexpected dysfunctional situations due to its (current) instability. Sometimes, virtual devices cannot be created or information is lost during device manipulation. Network limitations and synchronization issues were also present due to the size/amount of data that needs to be transmitted from the sensors to the virtual world.

However, we believe that these limitations can be overcome by (1) implementing better ways to transmit and process real data and (2) work around the virtual world limitations by considering other more flexible and configurable alternatives such as open source projects. Currently, we are also working on porting our solution into an open source virtual world implementation called OpenSimulator (<http://www.opensimulator.org>), which gives us the possibility not only to overcome limitations of SL, but also to enhance the current functionality of our framework, thus providing users with more efficient and effective ways to connect their real devices.

7 Conclusions

In this paper, we advocate an approach to virtual co-presence and collaboration based on the SL virtual world. SL

supports co-presence by synchronous multi-modal avatar communication, and collaboration via shared context and in-world object control. Our focus is on facilitating environmental studies in the agricultural field for common users, rather than computer experts. For this purpose, we developed the TWM framework, which manages the communication and interfaces between real devices and its virtual counterpart devices. Using the TWM, even non-programmers can easily set up a collaborative platform for visualization of real-world data and control real devices from within the virtual world. Visitors of our SL island may experience data from the NARC field server and control the Shinshu field server by following the links from our project web site (GlobalLab 2008–2009).

In our pilot study, we investigated a collaborative scenario in the agricultural field. Results indicate a high level of co-presence achieved in the scenario. Informal comments were generally positive.

While interaction between users happens mostly via text chat, the open nature of the virtual world client technology allows for the development of alternative client programs that could, in principle, handle richer, multi-modal types of interaction. Currently, the official SL client software only supports text and voice communication, but other data protocols could be implemented when supported by SL.

The only ‘programming’ required to use TWM is the specification of the DDL configuration file. The creation of this file requires some expertise in SL terminology and XML, which can be easily managed by most users. Having the DDL file correctly described, TWM eliminates the need to code the data formatting and synchronization, which are usually the most challenging tasks. This feature of the TWM contributes to the generic characteristic of the framework, i.e. the TWM can be applied to any kind of streamed data that can be visually represented as a virtual object (e.g. to display object positions in 3D map representations for positioning systems or to display weather variables as alterations of a virtual object’s physical properties such as color and lighting for weather report systems).

Since the TWM is built on SL technology, significant architectural changes of SL might affect its functioning. However, our component-oriented design guarantees that we would be able to easily adapt our connectivity component and thus be ready to support other types of visualization paradigms. On the other hand, open source alternatives such as OpenSimulator (2009), which can replicate the server functionality provided by SL, are getting more stable and increasingly popular. Since they offer additional degrees of control for more robust and efficient solutions in virtual worlds, we are also considering to port the TWM to this platform and increase its performance and functionality.

As a future enhancement of our framework, we are currently working on a definition of composite devices (i.e. devices that are constituted by two or more real devices) using DDL. This will allow users to define complex visualization schemas for data that are composed of different types of information, e.g. a virtual smartphone, which could be composed of a screen for image display, virtual buttons for manipulation, and an accelerometer for signalling movement.

Acknowledgments This research was partly supported by a “Grand Challenge” Grant from NII, Tokyo. We would like to thank Kazuki Kobayashi and Yasunori Saito from Shinshu University and the NARC research group for their kind support.

References

- Bainbridge WS (2007) The scientific research potential of virtual worlds. *Science* 317:472–476
- Boellstorff T (2008) *Coming of age in Second Life: an anthropologist explores the virtually human*. Princeton University Press, Princeton, NJ
- Boulos MNK, Hetherington L, Wheeler S (2007) Second life: an overview of the potential of 3D virtual worlds in medical and health education. *Health Inform Libr J* 24(4):233–245
- Brandherm B, Ullrich S, Prendinger H (2008) Simulation of sensor-based tracking in second life. In: AAMAS ’08: proceedings of the 7th international joint conference on autonomous agents and multiagent systems, International Foundation for Autonomous Agents and Multiagent Systems, pp 1689–1690
- Bray DA, Konsynski BR (2007) Virtual worlds: multi-disciplinary research opportunities. *SIGMIS Database* 38(4):17–25
- Bryson S (1996) Virtual reality in scientific visualization. *Commun ACM* 39(5):62–71
- Casanueva JS, Blake EH (2001) The effects of avatars on copresence in a collaborative virtual environment. Technical Report CS010200. Department of Computer Science, University of Cape Town, South Africa
- Du K, Sun Z, Han H, Liu S (2008) Development of a web-based wireless telemonitoring system for agro-environment. In: International Federation for Information Processing (IFIP), Springer, Boston, pp 799–807
- Eysenbach G (2008) Medicine 2.0: social networking, collaboration, participation, apomediation, and openness. *J Med Internet Res* 10(3):e22. <http://www.jmir.org/2008/3/e22/>
- Funding A (2007) Real life control panel for Second Life. <http://channel3b.wordpress.com/2007/01/24/real-life-control-panel-for-second-life>
- GlobalLab (2008–2009) GlobalLab project website. <http://www.prendingerlab.net/globallab/>
- Ichikawa (2009) Yamani Orchard, Ichikawa. <http://fsds.dc.affrc.go.jp/data4/Ichikawa01/>
- Kobayashi K (2009) Shinshu University’s field server. <http://www.cs.shinshu-u.ac.jp/~kkobayashi/ja/index.php?fs-obuse>
- Leigh J, Johnson AE (1996) Supporting transcontinental collaborative work in persistent virtual environments. *Comput Graphics Appl IEEE* 16(4):47–51
- libsecondlife (2007) libsecondlife project. <http://www.libsecondlife.org>
- Lifton J, Feldmeier M, Ono Y, Lewis C, Paradiso JA (2007) A platform for ubiquitous sensor deployment in occupational and domestic environments. In: IPSN ’07: proceedings of the 6th

- international conference on information processing in sensor networks, ACM, New York, NY, pp 119–127
- Lombard M, Ditton T (1997) At the heart of it all: the concept of presence. *J Comput-Mediated Commun* 3(2). <http://jcmc.indiana.edu/vol3/issue2/lombard.html>
- Mangan J (1997) Participatory pest analysis. In: PLA notes, 28
- Milgram P, Takemura H, Utsumi A, Kishino F (1994) Augmented reality: a class of displays on the reality–virtuality continuum. In: Proceedings of SPIE, vol 2351. Telemanipulator and Telepresence Technologies, Boston, pp 282–292
- Ogi T, Tamagawa K, Yamada T, Hirose M (1999) Collaborative scientific visualization in networked immersive virtual environment. In: IEEE international conference on systems, man, and cybernetics (SMC '99), pp VI-87–VI-91
- OpenMetaverse (2009) Virtual worlds connection library. <http://www.opensimulator.com/projects/libopenmetaverse>
- OpenSimulator (2009) Virtual worlds server technology. <http://opensimulator.org>
- SensorML (2009) Sensor model language. <http://www.opengeospatial.org/standards/sensorml>
- Valin S, Francu A, Trefftz H, Marsic I (2001) Sharing viewpoints in collaborative virtual environments. In: Proceedings of the 34th annual Hawaii international conference on system sciences
- von Kapri A, Ullrich S, Brandherm B, Prendinger H (2009) Global lab: an interaction, simulation, and experimentation platform based on “Second Life” and “OpenSimulator” (demo paper). In: Proceedings Pacific-Rim symposium on image and video technology (PSIVT '09)
- Weaver AC, Morrison BB (2008) Social networking. *Computer* 41(2):97–100
- Weber A, Rufer-Bach K, Platel R (2007) Creating your world: the official guide to advanced content creation for second life. Wiley, Indianapolis
- Zhao S (2003) Toward a taxonomy of copresence. *Presence: Teleoper Virtual Environ* 12(5):445–455