SIROS: A Framework for Human-Robot Interaction Research in Virtual Worlds

Antoine Raux
Honda Research Institute USA
Mountain View, CA, USA.
araux@hra.com

Introduction

In this presentation, we will demonstrate SIROS, a software toolkit that supports the development of 3D virtual-world based applications akin to online role-playing games, for the purpose of analyzing situated human-human and human-robot spoken dialog.

Simulated Environments for Human-Robot Interaction Research

PC-based virtual world simulations provide a light weight solution to perform Human-Robot Interaction research on robots that are still in early stages of development, or even purely hypothetical robots. Researchers can use simulators to collect data to build and evaluate interaction models at the same time as core components of the real-world robot are built and integrated. Once the real robot becomes robust enough, the models trained on simulators can be applied for further experiments.

Byron and Fosler-Lussier (2006), Orkin and Roy (2007), and Koulouri and Lauria (2009) present research conducted on such simulated worlds. However, in these prior systems, the situation does not dynamically change (besides the human-controlled character), making these systems unsuitable to the study of the dynamic aspects of situated interaction.

The SIROS Framework

Siros (**Si**tuated **Ro**bot **S**imulator) is a new client/server architecture developed at Honda Research Institute USA to conduct human-robot interaction research focusing on spoken dialog, using virtual worlds. Siros is similar to the architectures used by certain online video games. The server's role is to manage the virtual world and broadcast world updates to all clients so that they can be rendered to the human participants. The server receives commands from the Operator client (robot moves), runs the simulated customers according to the scenarios, and maintains the timer and the score. Anytime the trajectory of an entity (robot, customer, object) changes, the server broadcasts the related information, including entity location, orientation, speed, ... to all clients.

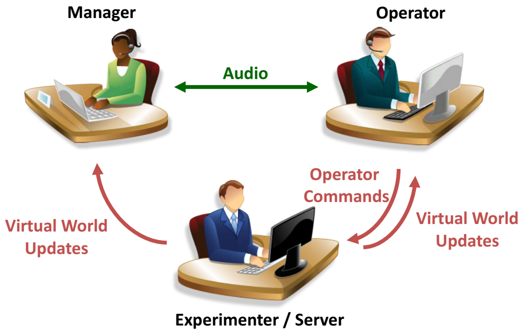


Figure 1: The Siros client/server architecture of the Konbini system.

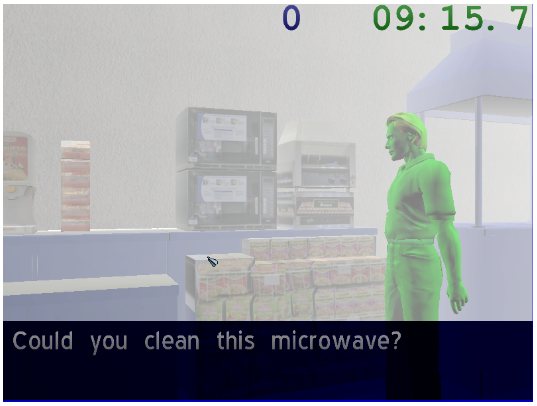
Clients are in charge of rendering a given view of the virtual world. 3D rendering is performed by the open source Ogre 3D engine (Open Source 3D Graphics Engine 2010). In addition, clients handle all required user interaction such as robot control and mouse-based object selection. Most importantly, clients log all messages and audio signals in a synchronized fashion, allowing analysis of the interaction between spoken content, user actions, and virtual world events.

Example Application: The Konbini System

Domain

In this system, two human participants collaborate in order to perform certain tasks pertaining to the management of a small convenience store in a virtual world. The two participants sit in different rooms, both facing a computer that presents a view of the virtual store. One of the participants, the Operator (O) controls a (simulated) humanoid robot whose role is to answer all customer requests. The other participant plays the role of a remote Manager (M) who sees the whole store from above but can only interact with O through speech.

Figures 2 and 3 show the Operator and Manager views. M can see the whole store at any time, including how many customers there are and where they are. In addition, M knows when a particular customer has a request because the customer's character starts blinking. M's role is then to guide O towards the customers needing attention.



Could you clean this microwave?

Figure 2: The Operator View.

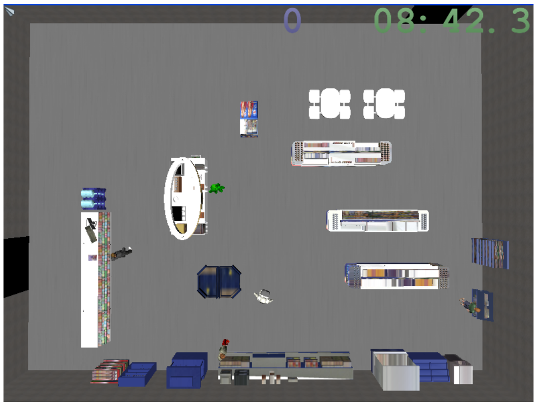


Figure 3: The Manager View.

On the other hand, O sees the world through the ”eyes” of the robot and can act upon the virtual world by clicking on certain object such as items on the counter (to check them out), machines in the store (to repair them when needed), and various objects littering the floor (to clean them up).

When the session begins, customers start entering the store at random intervals. Each customer follows one of 14 predefined scenarios. Scenarios represent the customer’s moves in terms of fixed way points. As a simplification, we did not implement any reactive path planning. Rather, the experimenter, sitting in a different room than either subject has the ability to temporarily take control of any customer to make them avoid obstacles.

The Konbini Corpus

Using the system described above, we collected 18 10-minute sessions from 18 participants randomly paired into teams. In addition to the audio recordings and event logs, we manually annotated all sessions for operator and manager utterances (orthographic transcription and dialog acts). Raux and Nakano (2010), provides a detailed analysis of the dynamics of interaction focusing on correction utterances in this corpus.

Demo

The presentation will feature live demonstration of the Konbini system, as well as videos extracts from the Konbini corpus with Anvil annotations.

Acknowledgments

I would like to thank my colleague Mikio Nakano for many helpful discussions and feedback during the development of Siros and the Konbini system.

References

Byron, D. K., and Fosler-Lussier, E. 2006. The OSU Quake 2004 corpus of two-party situated problem-solving dialogs. In *Proc. Fifth International Conference on Language Resources and Evaluation (LREC 2006)*.

Koulouri, T., and Lauria, S. 2009. Exploring miscommunication and collaborative behaviour in human-robot interaction. In *Proc. of SIGDIAL*, 111–119.

Open Source 3D Graphics Engine, O. 2010. http://www.ogre3d.org.

Orkin, J., and Roy, D. 2007. The restaurant game: Learning social behavior and language from thousands of players online. *Journal of Game Development* 3(1):39–60.

Raux, A., and Nakano, M. 2010. The dynamics of corrections in situated interaction. In *SIGdial 2010*.