Inquiry: The University of Arkansas Undergraduate Research Journal

Volume 11 Article 13

Fall 2010

Beyond the Pie: Communicating with Smart Objects Using Menu-Based Natural Language Interfaces

Tanmaya Kumar
University of Arkansas, Fayetteville

BEYOND THE PIE: COMMUNICATING WITH SMART OBJECTS USING MENU-BASED NATURAL LANGUAGE INTERFACES

By Tanmaya Kumar

Department of Computer Science and Computer Engineering

Faculty Mentor: Craig Thompson

Department of Computer Science and Computer Engineering

Abstract

In both real and 3D virtual worlds, people and avatars (representations of people) need to be able to communicate with things around them. Without guidance, however, people cannot use the language that the things can understand. The goal of our research is to extend the 3D virtual world Second Life® to better model pervasive computing and overcome the boundaries of communication. The aim of this paper is to show how to build a dynamic menu-based user interface that enables humans to communicate with model entities. The focus is the applicability of object-specific grammars associated with things (objects in the real and virtual worlds) and a GUI consisting of cascaded menus to guide people in "talking to" things. This paper discusses the prototype model of a new virtual controller that takes us closer to the ultimate goal – a system that extends the Second Life user interface so that people can task robots using a menu interface

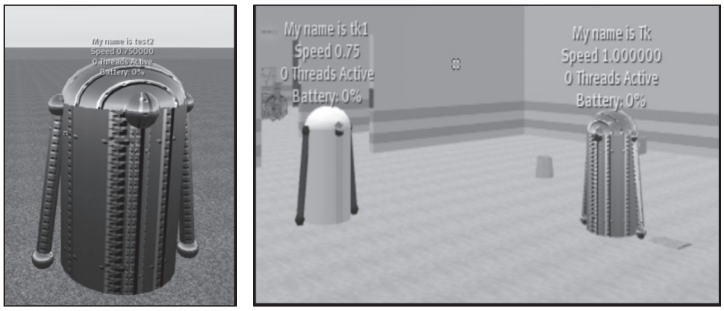

Figure 1. The bots on University of Arkansas Island in Second Life, a 3D Virtual World.

Context

This report is associated with the "Everything is Alive" project at the University of Arkansas, which is exploring pervasive computing both in real-world RFID applications [1] and in virtual worlds, especially Second Life [2] and Open Simulator.

Problem

People use natural languages to talk to other people. Researchers have been trying to develop natural language interfaces (NLIs) to talk to databases, for example, for the past 40 years, but success has been limited. It is currently difficult to impossible for people to communicate and converse using NLI with most non-human things around them (e.g., chairs, thermostats, pets, blood pressure machines, and forklifts). A recognized reason is the habitability problem [1]: humans overshoot and also undershoot a system's ability to understand human language. Overshooting means that people use language that the system fails to comprehend, so the system is unable to respond to the command appropriately. Undershooting means that people make the system execute very trivial tasks and do not realize the capabilities of the system, thus failing to use many powerful features of the system.

Another major issue with objects around us is that they do not explicitly know their own identity or type. The concept of ontologies is absent ("I am a unique chair"), and the objects have no way to associate additional information with themselves (*"I am owned by Tanmaya"* or *"I am a light switch that has been turned on 313 times this year"*).

People do not simply want a way to talk to real-world objects. In virtual worlds, in-world objects may have associated information and scripts, but the ability to extract the information or manipulate the object may rest solely in the head of the object developer. No avatar passing by can learn the command language of the object and interact with it. To aid the user, Second Life offers the PIE user interface, which a user can access by selecting an object and giving generic commands such as *sit, take, copy,* and *buy*. However, none of these commands are object specific, nor do they allow the user to manipulate the special capabilities the object may have. For example, thermostats do not have their own object type specific commands.

Objective

The aim of this research was to enable human-object interaction in both real and virtual worlds by providing a GUI interface that can be parameterized by the physical properties of everyday objects. In the initial implementation, interactions were limited to one type of entity. In the Second Life 3D virtual world, robots with command language modules are available, so the form-based natural language interface was tested by commanding and querying the robots.

As mentioned above, Second Life currently employs a pie-shaped interface that allows the user to select from a "fixed and limited" array of generic verbs that allows the avatar to perform very basic actions such as sitting and touching objects. Even though the pie can cascade to give a small number of additional commands, the pie commands are not object-specific. What is needed is a way to extend this user interface to communicate with things in a thing-specific language. In the current work, the target is the Second Life robots on the University Island.

Approach

A series of Form Based Graphical User Interfaces (GUI's) was used which provides a common way for humans to communicate with computer-based objects. A complementary alternative is Menu Based Natural Language Interfaces (MBNLI'S), which provide sequential command completion menus [3] similar to drop-down menus. Both alternatives provide a way to solve the habitability problem since both provide a way to display all and only the legal commands a system can handle. Instead of "creating" an unsupported query, as is done in conventional NLI, by using GUI or MBNLI, human users can "recognize" the command they mean to formulate while creating an appropriate string of commands with a command builder. This method also enables humans to see commands that they might not have known about. That is, humans are guided to rendezvous with the capabilities of the system, thus eliminating the chance of a user undershooting or overshooting a system's capabilities.

Progress

We developed a prototype next-generation PIE interface for Second Life that uses a combination MBNLI-GUI to enable humans to communicate with specific things. In our initial implementation, we limited interactions to one type of entity: robots [5]. A student at University of Arkansas, Nick Farrer, had previously developed a Robot Assembly Language that provided chat-based commands in Second Life to control a fleet of robots that can go from location to location following way points and can pick up, carry, and put down objects.

In order to get a PIE that operated in both Second Life and OpenSim, we developed our PIE code outside both environments so that it could overlay as an external application on top of those browser-clients. We developed object-specific grammars such as the one shown below for robots. If the user clicks on a robot, the grammar commands for the robot are interpreted and displayed in a cascade on the menu. At the end of a PIE command sequence such as "Robot – Pickup – the ball", the command is translated into a command in the Robot Command Language, transmitted over to the robot in second life via an HTTP callback, and then executed.

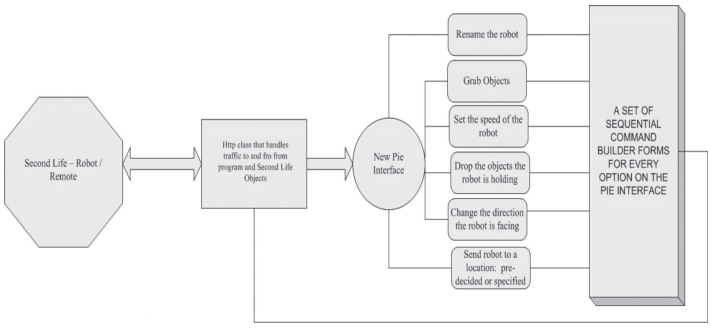

Figure 2. The program design and interaction paradigm. The change from one pie to another is seamless, as the new pie has been integrated into a small script that corresponds to the main program running on a stand-alone system via an HTTP callback and is activated by a single left mouse click on the object the avatar is interested in. The new Dynamic PIE 2.0 is shown below against the pie menu provided by Second Life®.

The change from one pie to another is seamless, as the new pie has been integrated into a small script that corresponds to the main program running on a stand-alone system via an HTTP callback and is activated by a single left mouse click on the object the avatar is interested in. The new Dynamic PIE 2.0 is shown below against the pie menu provided by Second Life®.

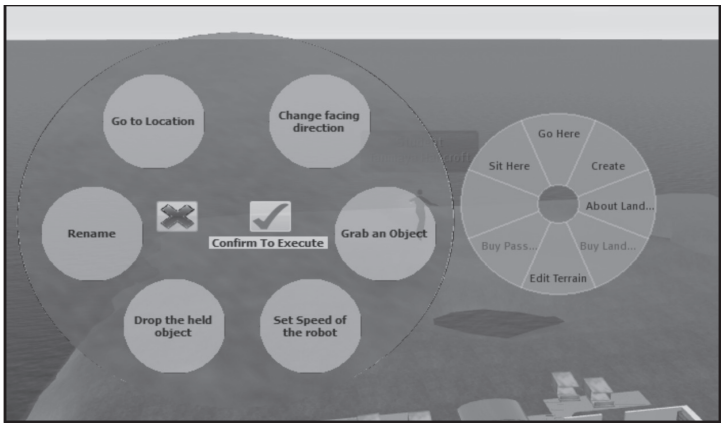

Figure 3. The new pie interface (left) [2] vs. the one provided by Second Life (right).

Implementation and Hurdles

The primary aim of the project was to develop an interface to facilitate the interaction. The interface was built entirely outside Second Life® using Visual C#. It interfaces with objects inside Second Life® using an HTTP callback class which ferries commands and requests to and from the target object: in this case, the robots and the soft controller/transmitter on the avatar's hand. The stand-alone nature of the PIE 2.0 is not accidental; it was developed as a plug-in to Second Life®, OpenSim, and all supported virtual worlds to support extreme portability and to avoid the legal software licensing issues afflicting Second Life and Open Simulator. [6]

Sample Command Building Process

The new pie interface works by using a pre-programmed set of forms as interfaces for the command development. Normally, the bots function by listening on the common chat channel, "Channel 0," for commands being issued to them by the user (avatar). The commands are all typeset and are hard to memorize, formulate, and reproduce at the desired instant. The following is a sample workflow for a worker robot collecting a clock from a shelf and moving it to the loading dock

1) SampleBot rename to MyBot (Renames robot to MyBot)
2) MyBot wp shelf2 (Sends robot to waypoint named shelf2)
3) MyBot grab nearest Clock (Grab clock from the shelf)
4) MyBot wp loading (Sends robot to loading dock)
5) MyBot drop (Robot drops clock at the loading dock)

The following figure shows how some of these commands can be built in the new interface without having to have these command steps memorized.

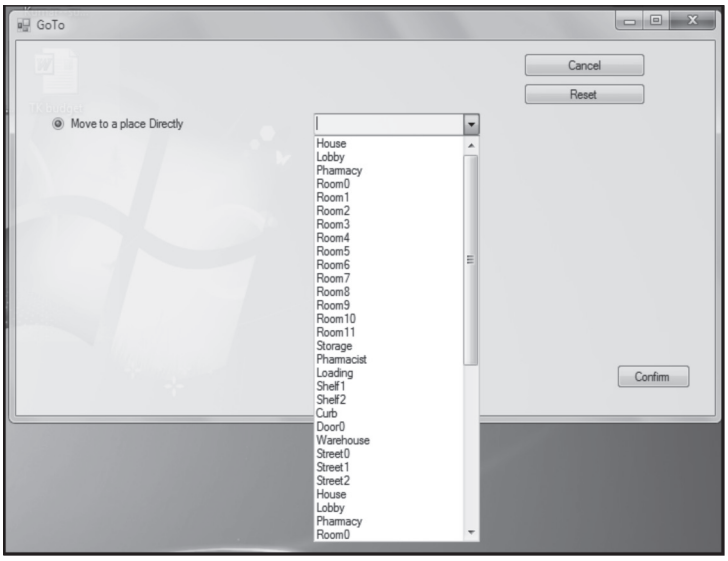

Figure 4. An instance of the Go to Location Command builder and list of way-points.

Future Work

The ability of humans to communicate with things is a significant step towards a smart world (e.g., smart homes, smart buildings, and smart hospitals) where many or all objects are network objects with an identity and the ability to communicate (wired or wirelessly) with other objects or with humans. The results of the current research enhance our understanding of the working of the Second Life PIE in detail. It was a prototype hardcoded to control the Robots on "University of Arkansas" island in Second Life. Even then, the interface did not cover all of the command language. Most other Second Life objects still lack the ability to understand their own type and any super- and sub-classes, which indicates another place to begin. In addition, the grammars are not yet dynamically loaded into the PIE. While there is considerable work ahead, a set of problems to solve has been isolated.

Some requirements for future work will include the development of objects that can identify themselves as belonging to a parent class, such as a chair knowing it is a type of furniture. In addition, there is a need for an API that recognizes the type of object and recalls the particular set of grammar rules that apply to the object, possibly even differentiating the availability of grammar rules on basis of the user. Last but not the least, it is necessary to include the concept of object-specific grammars and the need for developing grammars for every kind of real-world thing. More specifically, the following goals need to be achieved:

1) Consider other Second Life objects such as the objects of a smart home. If grammars can be developed for everything and put in a remote cloud to be accessible to the users, and if the users can download them on their smart phones, the boundary of communication between a people and their surroundings can be breached. Users can then, in a simple sense, talk to anything around them wherever they are. (Fig. 6)
2) Build a more advanced interface that can handle complex commands using a grammar-based control structure, complex translations, and ontology structures. (Fig. 7)

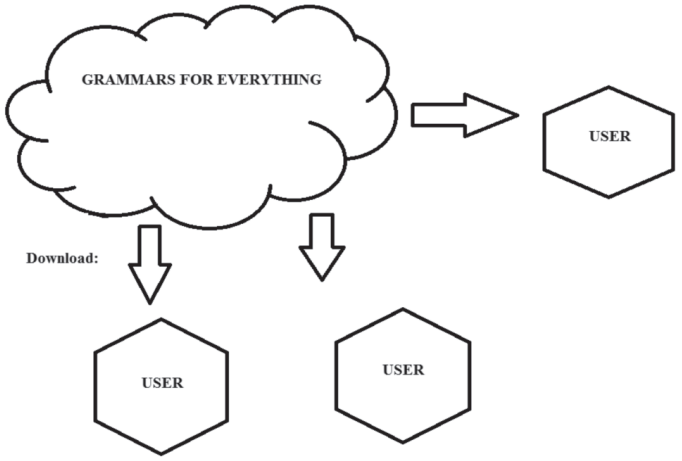

Figure 6. The grammar cloud concept

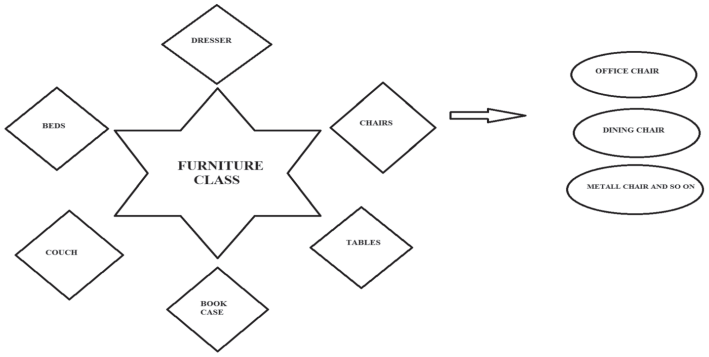

Figure 7. Ontology Architecture

3) Develop role-based interfaces such as those specific to a particular task (e.g., a wood-cutting machine for a car penter or automotive tools for a mechanic).
4) Develop mechanisms to control and limit access by users by differentiating authorized and unauthorized access or by attaching an inventory of objects to a particular avatar so only that avatar can access them. This feature is similar to the act of owning and commanding objects in one's own home.

Finally, if smart grammars work in Second Life, they can be made to work in the real world. If every real-world object has an RFID tag that identifies it and if every smart phone can read RFID tags (which will happen when RFID becomes a consumer-level technology), then consumers can shop or walk around and use their smart phone to communicate with things around them (e.g., canned goods, the thermostat, their cat) using technology similar to the technology being designed.

Potential Impact

If we can determine the kinds of interfaces an object can possess, develop a synthetic grammar for the commands and replies for the object, and extend the communication interface to support interaction, it will make it possible for humans to interact with objects. A similar approach can be used for people using soft controllers (smart phones) to communicate with everyday objects

in the real world. If every real world object has an RFID tag that indicates the object's individual ID, a smart phone with an RFID reader can communicate this information to a remote ontology on the web to download an interface that will let a consumer talk to the thing. If it becomes a standard (optional) protocol to define such interfaces for all things, then anyone anywhere can communicate with any tagged thing.

References

[1] Everything is Alive project website (aka Modeling Healthcare in a Virtual World), http://vw.ddns.uark.edu
[2] Second Life website, http://www.secondlife.com
[3] C. Thompson, P. Pazandak, H. Tennant, "Talk to your Semantic Web," IEEE Internet Computing, November-December 2005. http://csce.uark.edu/~cwt/DOCS/2005-11--PAPER--IEEE-Internet-Computing--Talk-to-your-Semantic-Web.pdf
[4] T. Kumar, "Enabling Communication between Humans and Inanimate Objects in a Virtual World," Final Report, Undergraduate Research Project, Honors College, University of Arkansas, Fall 2009. http://vw.ddns.uark.edu/content/2009-10--Honors-College-Research-Grant--Final-Report--Tanmaya-Kumar.doc
[5] N. Farrer, "Second Life Robot Command Language," Tech Report, CSCE Department, University of Arkansas, Fayetteville, AR, 22 February 2009. http://vw.ddns.uark.edu/content/2009-02--SL-Robot-Command-Language-v0--Nicholas-Farrer.doc
[6] "Open Sim"; Licensing Issues, website: http://www.opensimulator.org/wiki/Contributions_Policy

Mentor Comments: Craig Thompson points out that, unlike many freshmen, Tanmaya Kumar began his studies at the U of A with a clear commitment to the investigation of artificial intelligence. As a result, he has been very productive in his time here.

From the beginning of his freshman year in Fall 2008, Tanmaya knew he wanted to major in Computer Science and do research in the area of artificial intelligence. Funded by two UA Honors College Undergraduate Research Grants, he joined my Everything is Alive (EiA) research project which is focused on pervasive computing. Tanmaya was particularly interested in how humans can communicate with real world things. The thesis is that in the smart world of the future, every object will have identity, the ability to communicate with other objects – and we need a way so that people can communicate with objects to command and query them. Tanmaya built on my earlier research on menu-based natural language interfaces that guide a human so they know what domain restricted language they can use to talk to a specific kind of thing. Furthermore, he observed that the Second Life user interface (a menu in the shape of a pie that provides a user choices when they click on an object) is extremely limited and generic. He decided to extend the Second Life pie menu so the user could provide their own object-specific commands. He demonstrated his new pie interface tasking robots with commands like – Robot, pick up the book and take it to the library. His work is valuable not only in the virtual world but can also be applied directly to real world objects. If we attach RFID tags to real objects and add RFID readers to cell phones, we will be able to upload user interfaces to cell phones so humans can use these smart phones as truly universal remotes to communicate with tagged things around them. To document his work, Tanmaya wrote the paper ""Beyond the Pie: Communicating with Smart Objects using Menu-Based Natural Language Interfaces"" for the X10 Workshop on Extensible Virtual Worlds (http://vw.ddns.uark.edu/X10, March 29-30, 2010). During the summer 2010, Tanmaya is extending his initial work into a project we are calling "My Immortal Avatar" (http://vw.ddns.uark.edu/X10/content/APPLICATION--My-Immortal-Avatar--Kumar-Thompson.pdf¬) which aims at using an avatar chatbot to model an individual human's looks and memories - a new biography genre