

Using Non-Player Characters as Tutors in Virtual Environments

Clinton Jeffery

Keywords (max 5): e.g. Second Life, World of Warcraft, non-player characters

Abstract:

Massively multiuser online games such as World of Warcraft employ computer controlled non-player characters and quest activities extensively in training/tutoring capacities. This approach is very effective and popular, incorporating active learning, incremental progress, and creative repetition. This paper explores ways to exploit this model in educational virtual environments. Non-player characters in such environments require a knowledge model, a dialogue model, and a user performance model, in addition to any physical and behavioral traits necessary to make them interesting and credible members of the environment.

A first experiment with these ideas is an explicit attempt to embed a World of Warcraft-like tutoring non-player character in Second Life. Second Life supports virtual objects with scripting, but this facility is not particularly tailored to non-player characters or to education. With care, compatibly formatted web-based content, exercises, and quizzes are imported into Second Life, reducing the effort needed to create content.

A second experiment with these ideas is to ask: What should non-player character construction by end user educators look like? A set of tools and libraries designed to reduce the effort required to construct educational non-player characters within a custom educational virtual environment is described. The same educational content format is used in the first and second experiments. The tools or underlying ideas may be profitably adapted by other educational virtual environment construction systems.

Introduction

The spectacular success of Massively Multi-user Online (MMO) games has led to a large amount of interest in educational multi-user virtual environments. World of Warcraft (WoW) and similar games have demonstrated both the mass appeal and the potential of this genre. From Chinese Coca Cola to North American Toyota commercials, WoW has penetrated beyond the normal sphere of influence of computer games into popular culture. Organizations such as the U.S. Army are using custom games for recruiting and training purposes, and NASA has proposed an

educational MMO to use a space theme to promote interest in science and engineering education and careers.

As yet, no one has produced an educational MMO with a significant following. It was speculated at the NASA MMO workshop in April 2008 that the education software industry was not economically able to attempt a large-scale educational MMO. Academia itself has many groups eager to work on the problem, but lacking the resources to realize their ideas. MMO's take years and millions of dollars to develop. A few organizations such as Harvard have developed small-scale educational virtual environments by hiring game companies to write them atop proprietary game engines, but most educators that wish to work with MMOs find their way to a general-purpose MMO such as Second Life.

The focus of Second Life is user-created content. The landscape, buildings, virtual objects, and the appearance of the avatars are largely the result of end-user actions. Educational institutions have presences there for marketing purposes and on-line lectures and discussions. They are in the process of exploring the educational potential of Second Life, but the cost of building educational content (beyond virtual buildings and meeting places) is high. Importantly, Second Life does not directly support the quests and activities that are one of the main reasons to play MMO's, and are similar to the activities that are the mainstay of conventional education software. Similarly, Second Life has no model of experience or skills development which are a major motivator for users to play MMO games.

Second Life provides a scripting language by which virtual objects can incorporate behaviour, and access to external content via protocols such as HTTP. One can build anything with this framework. However, without support for modelling activities and user experience, the cost of developing substantial educational applications in Second Life is prohibitively high for most educators.

The project described in this design paper began not within Second Life but rather with a custom educational virtual environment for computer science distance education, called CVE. With the goal of enabling distant students to attend lectures and office hours, and do homework assignments and labs within the virtual environment, initial efforts focused on reproducing a local CS education environment, including 3D representations of two physical CS departments, avatars, text and voice chat, and interactive collaboration on common CS tasks of editing, compilation, execution and debugging. The initial project was placed on Source Forge for public access at cve.sourceforge.net, but its authors had to face the question: who cares? Perhaps some developers of Second Life projects have faced a similar dilemma: the virtual environment lacked the compelling aspects of discovery, quest activities, and experience levels and advancement which make MMO's special. It was decided to solve this problem for both the CVE and Second Life virtual environments, in order to learn from the comparison. The central building blocks for the experiment are the computer-controlled non-player characters (NPCs), who serve as tutors and record keepers for users' accomplishments.

Non Player Character Tutors

In Role Playing Games and MMO's, computer-controlled NPCs play a vital role as quest givers or assistants to the user who goes on an adventure. Fig. 1 shows the very recognizable yellow exclamations over NPC quest givers' heads in WoW. This marks NPCs with available quests as persons in the environment with whom the user has special reason to interact. The dialog on the left side of Fig. 1 shows a typical WoW quest, which is shown in response to a right-click on an NPC.

Quest activities are used to teach the game itself as well as to entertain. WoW quests do not resemble interaction with other player characters, which is through text and voice chat. WoW features about 10 types of quests, requiring the user to perform various tasks, such as combat, exploration, delivery, and gathering or manufacture of virtual items. Other MMO's provide additional kinds of quests. By analogy to Second Life's user-defined virtual objects, future educational MMO's will need end user teacher-defined quest activities. This paper describes an approach to providing that capability in a multi-platform manner, proposing the development of a Multi-Platform NPC (MPNPC) standard, focusing on user created quest activities. Although MPNPC claims platform portability, this paper focuses on providing activities usable on Second Life and CVE, not proprietary game systems such as WoW.



Figure 1: A non-player character and quest activity in World of Warcraft

A MPNPC tutor is created much as is a regular (human-controlled) character. End users can utilize their regular account or an auxiliary account to create an NPC character, adding quests and activities available to other users. When the user is logged off, their avatar is still present on the system, controlled from a remote NPC server machine and functioning as a quest-giving NPC and a virtual secretary,

interacting with other users as instructed by the player. Many educators will choose to create multiple “characters” playing specific educational roles, giving them an artificial personality and a set of activities for other users, and leaving them under the control of the computer full-time. There is a reasonable question of whether these characters should be replicated simultaneously in each virtual world, or whether they should exist in one world at a time and move explicitly from world to world. While MPNPC supports both models, it is desirable in most cases to instantiate NPC’s in each virtual world in order to share educational content across platforms.

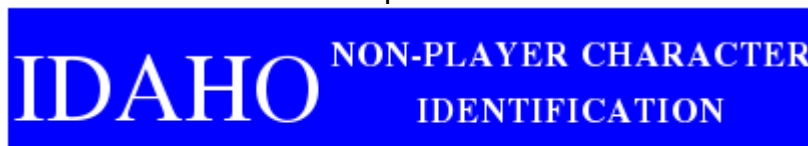
NPC Profiles

NPC’s 3D appearance is supplied via an avatar creation tool, a task that is not considered in this paper. The more important task of specifying the interactive and tutoring behaviour of an MPNPC consists of defining the knowledge model, dialogue model, and behaviour model for that NPC. These are specified in a *profile* file which is pulled in via HTTP at NPC login time. The profile is used by an *NPC agent*, which is a thread or process that plays the role of a client on the virtual world server.

An NPC profile is a file containing NPC details in simple HTML (for handwritten profiles) or XML (for machine generated profiles) format. The intent is that an NPC Profile can be created and maintained as a webpage. An NPC profile contains the following sections. In HTML they would each be given in a named anchor tag. Although a friendly graphical wizard for creating profiles is available, many NPC’s can be created manually by copying a template and changing the content details.

id

An "ID card" presentation of the NPC, suitable for use on an "inspect details" operation in a game. The id provides an image, name, and other basic attributes. Here is an example:



EXPIRES 12/31/08



Name: Tux

Title: Linux/UNIX Trainer

Home: JEB 026

cve://uidaho

Visits: SecondLife

Dob: 8/8/08

 knowledge

The knowledge model is a specification of the knowledge that this NPC has to offer. This is a bulleted list of named links to quests.

behavior

The behavior is model a specification of this NPC's active (e.g. mobile) behavior. Values include stationary, wanderer, and companion.

dialogue

The dialog model is a specification of this NPC's dialogue capability in AIML

avatar

A specification of this NPC's avatar (link to 3d model file, scale, textures)

Knowledge Model

Real Knowledge Representation will be vital in making smart NPC's of the future, but for MPNPC's, there are two types of knowledge that matter: what quests they can offer, and what they remember about other characters' experience from past quests and in-game events. The former is almost static, refined occasionally by the NPC's creator. The latter is dynamic and should occur automatically during game play.

An MPNPC's quest activities are the primary mechanism for tutorial learning. The format of the actual quest activities is intended to resemble UML use case descriptions (Booch, 1998). Since the virtual environment will not help track quest completion, the kinds of steps available are limited to those that can be observed by the NPC(s) interacting with the user in the supported virtual environments. The following is an example quest from the domain of computing:

```
Name
    ls
Summary
    Learn the basics of the ls(1) command.
Requires
    Files, Directories
Steps
    1. Read the UNIX manual page for ls(1).
    2. Pass a quiz on ls command line options.
    3. Demonstrate "ls" for Tux.
Rewards
    UNIX: 1
Quiz (2/2 to pass)
    How can you get a long listing that shows file
    permissions and size?
    > ls -l
    How can you list all files in all subdirectories?
    > ls -R
Demo (2/2 to pass)
    Show me a simple listing of the root directory.
    > ls /
    Show me a listing of the current directory, sorted
    by the time each file was last accessed?
    > ls -t
```

The main differences between a quest and a use case description are that a quest may contain auxiliary content (such as quizzes and demonstrations) that are used to measure completion of the quest steps, and a quest lists rewards for completion, if any. Quizzes and demonstrations will often need to be external references to pools of questions. The difference between quizzes and demonstrations is that a quiz is delivered and answers interpreted by an NPC agent directly, while a demonstration involves an in-world interaction (in this case, a session with a tutorial UNIX command-line shell) that is monitored by an NPC agent. Evaluation of deeper

understanding may require offline human evaluation, or fall outside the realm of what an NPC Tutor can reasonably perform.

Dialogue Model

There is a semantic gap between WoW-style NPC interaction, and the text and voice chat used by human-controlled player characters. Some MMO's such as Everquest reduce the gap by forcing NPC interaction through the chat channel, with greater realism but more user frustration when an NPC refuses to give up a desired response to a player's guesses. Although WoW-style interaction is easy, chat-based interaction is more "portable" across virtual worlds. An MPNPC's dialogue model consists of a chat script written in AIML (Wallace, 2005), embedded with offers to undertake available tutorial quest activities. The AIML chat scripts provide rules which determine how the NPC will reply to questions initiated by a player.

Behaviour Model

The NPC behaviour model provides a set of rules for NPC movements and responses to external stimuli. Since most WoW quest-giving NPC's do no movement, the behaviour model is not considered extensively in this paper and only three values are described: stationary, wanderer, and companion. An NPC whose behaviour model is stationary does not move from a specified home location. A wanderer is an NPC that moves randomly within a prescribed domain. A companion NPC accompanies a player on a destination-based quest.

Experience and Reward via Peer Review

Players need to be motivated to perform quests, and their accomplishments need to be recognized. Each quest completed by each player from any virtual world needs to be rewarded and remembered. Although such "remembrance" could be associated with the specific NPC for whom the quest was completed, the MPNPC system relies on an external SQL database. Rewards in the form of virtual objects or clothing might be duplicated on multiple virtual worlds, while other rewards may be specific to a particular world (such as Linden dollars, or a new character ability).

The main kind of reward that matters to MPNPC's are the experience points in specific skills that enable a character to undertake more advanced quest activities. In the Is activity described above, two specific previous quests (tutorials on Files and on Directories) had to be completed before the NPC would offer the Is quest. Completing the Is quest enables any quests that depend on it specifically, and also awards a point of general UNIX experience.

When teachers create new tutorial quest activities, there are basic questions about them, including how fun they are, and their educational content. The fun is best evaluated by the quest completers, while the educational content is best evaluated by experts and other peer educators. Thus, when an activity is created its author can suggest its category and experience point value, but those entries are honoured only after the activity has been performed by others and rated.

Architecture Issues

The most interesting design consideration for NPC's in educational virtual environments turns out to be how much of the NPC appearance or behaviour is coded inside the virtual environment using its normal programming API's and

scripting mechanisms versus how much is coded externally via a separate program that communicates with the virtual environment.

In this project, it was initially anticipated that NPC's would be coded largely inside the virtual environment's normal programming API's, such as Linden's LSL scripting language. Portability needs overrode this intuition. The more MPNPC logic that can be coded outside the virtual world server, in the NPC process, the more of that logic can be shared across virtual worlds.

Second Life NPC's

Several interesting prior, related experiments have been conducted to add non-player characters to Second Life. We are not aware of existing characters that have been used to deliver tutorial quests.

Art Fossett's blog describes a non-player character created as a virtual object (Fossett, 2008). Making an object look humanoid is a challenge in Second Life, but can be accomplished using sculpted primitives, which are a restricted form of 3D model that graphic artists can produce with commercial grade tools at substantial effort. Fossett couples this humanoid-looking virtual object with an external "chat" program called a PandoraBot, which implements AIML chat and plays a role similar to the MPNPC dialogue model.

Doron Friedman et al (Friedman, 2007) built a Second Life NPC by taking an ordinary user avatar and attaching a virtual object (a ring) to it that turns the avatar into a puppet controlled by an external program. This NPC can move around the environment, albeit with very simple rules for essentially random movement.

The MPNPC Second Life implementation borrows ideas from both of these approaches, namely the AIML and the use of a user avatar rather than an attempt to construct a facsimile from virtual objects. As is the likely case for many other Second Life NPC projects, MPNPC is written using libopenmv (www.libsecondlife.org).

NPC's in the CVE Environment

The CVE virtual environment is primitive compared with Second Life, but its simplicity allows easy experimentation. CVE is written in Unicon (www.unicon.org), a very high level language, rather than a systems programming language. CVE avatars can be created from 3D models produced by tools such as 3D Studio Max and exported in Microsoft .x format. Fig. 2 shows some example models in the CVE environment.

Compared with Second Life, CVE features an integrated collaborative IDE that allows tutorial activities for a range of computing topics. Besides shell commands illustrated in the "Is" example earlier, these include editing, compilation, execution, debugging, and testing activities for C, C++, Java, and Unicon. Different NPC's teach different subjects and have different personalities embodied in their AIML scripts. In order to offer these CVE-native tutorial activities in Second Life, the NPC agents in Second Life must provide the required interactive demonstration facilities themselves, or accept file submissions of captured sessions conducted outside the virtual world.

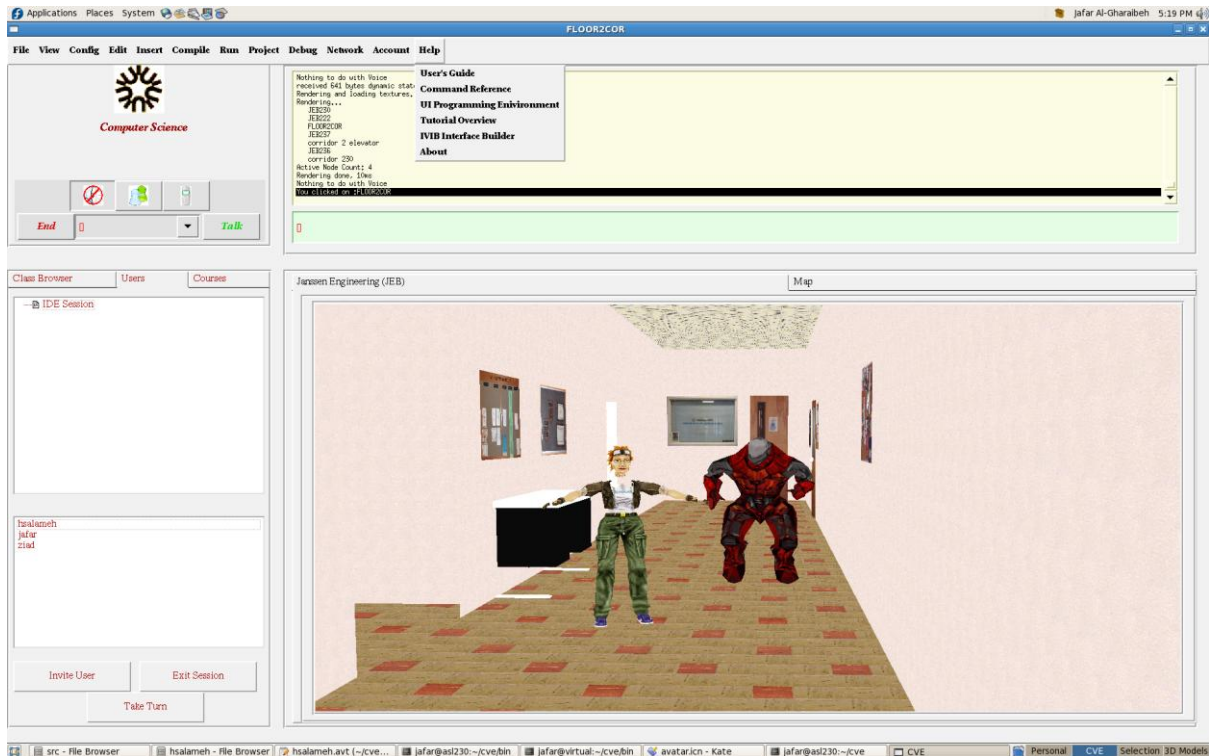


Figure 2: CVE NPC's 3d models are loaded from .x files

Conclusions

This paper introduces MPNPC, an experiment in multi-platform non-player character architecture. The goal of the experiment was to provide World-of-Warcraft style quests for the purpose of encoding educational content. The MPNPC architecture provides rudimentary NPCs that offer educational quest activities to users across virtual worlds. It seems probable that the quests that are the most fun will involve interactions that are specific to a particular virtual environment. However, judging from WoW's quest mix, a large percentage of quests can be world-independent. For tutorial NPC's they will likely be domain-dependent for a learning domain which may or may not have a virtual world embodiment. Without such direct virtual world embodiment of the material being taught, the success of tutoring is defined mainly by the NPC's ability to convey the material via its scripts.

Given the availability of interesting NPC's that live in multiple worlds, significant challenges remain, such as providing rewards that provide increased character capabilities that function across multiple worlds. However, since most users will interact with MPNPC's through a single virtual world interface, the primary function of these tutorial NPC's is to enable content to be used in multiple environments.

Acknowledgements

Fig. 1 in this paper was provided by Jafar Al-Gharaibeh. The research was supported in part by NSF ATE grant number 0405072.

References

Booch, G., Rumbaugh, J., and Jacobson, I., *The Unified Modeling Language User Guide*, Addison-Wesley, 1998.

Fossett, A., *PandoraBot NPC*. Web page obtained September 2008 from <http://artfossett.blogspot.com/2007/07/pandorabot-npc.html>.

Friedman, D., Steed A., and Slater M., *Spatial Social Behavior in Second Life, Proc. Intelligent Virtual Agents LNAI 4722*, Pelachaud et al. (eds), pp 252-363, Paris France, September 2007.

Wallace, R., ed., *Artificial Intelligence Markup Language (AIML) Version 1.0.1*. Working draft obtained from <http://docs.aitools.org/aiml/spec/>.

Authors

Clinton Jeffery

Department of Computer Science, University of Idaho, Moscow ID 83844 USA

jeffery@uidaho.edu

Clinton Jeffery is an Associate Professor in Computer Science at the University of Idaho. He has researched in virtual worlds since 2004 and leads the CVE project on Source Forge. His name in Second Life is Jeffery Clinton, which he frequently hears incorrectly in real life.