Presence in Technologically Mediated Environments: A Research Framework

Chang Nam and Steve Johnson
Department of Industrial Engineering, Department of Computer Science & Computer Engineering
University of Arkansas, Fayetteville, AR 72701, U.S.A
{cnam@uark.edu, sajohns@uark.edu}

Abstract

The paper outlines a research framework that can serve as a test-bed to presence research. Few system architectures are still available for presence researchers to use as a test-bed in order to investigate research issues involved in perceived presence in technologically mediated environments. The framework allows researchers to manipulate and investigate system configurations and parameters (e.g., dynamically changing agent roles, autonomy level, etc.), as well as psychological and social factors, which may hinder or facilitate the user's perceived presence in multi-agent contexts. The framework is an adaptable multi-agent system that interfaced intelligent, autonomy-based agents to a human commander within a real-time simulation environment. The research framework developed in this paper would be able to provide a basis for more flexible and systematic study on presence in technologically mediated environments.

1. Introduction

Advances in immersive, interactive technology, combined with its increasing availability and quality, have resulted in a practical concern with the manner in which people interact with technologically mediated environments such as virtual environments, haptic systems, and 3D online games. Terms like presence [1, 2], telepresence [3], and virtual presence [4] are used interchangeably to describe the extent to which people perceive that they are actually present in the artificially created virtual environment. Many attempts have been made to define such experience - *the feeling of being there that is created by media technologies* - and identify its determinants. Researchers have often developed an experimental system for their presence studies. More recent studies, on the other hand, utilize commercial off-the-shelf systems (such as 3D online games) as a test-bed.

These systems tend to limit the research scope of a presence study, however, because they do not allow the researcher to easily modify system configuration in a way that is cohesive to the research goal(s). That is, there are few system architectures available for presence (and even copresence) researchers to use as a test-bed in order to investigate research issues involved in perceived presence in technologically mediated environments. In addition, little attention has been paid to the presence of the user who is interacting with intelligent, autonomy-based agents - as in adaptable multi-agent systems.

This paper aims to establish a research framework that allows researchers to manipulate system configurations (e.g., task type, agent characteristics, etc.) and parameters (e.g., dynamically changing agent roles, autonomy level, etc.), as well as psychological (e.g., personality, trust, etc.) and social factors (e.g., collective efficacy, etc.), which may hinder or facilitate the user's perceived presence in multi-agent contexts. The framework is an adaptable multi-agent system that interfaces intelligent, autonomy-based agents to a human commander within a real-time simulation environment. The research framework developed in this paper would be able to provide a basis for more flexible and systematic study on presence in technologically mediated environments.

The paper begins with an outline of the UT2003 game environment, which is a major system component of the framework. A development strategy to implement intelligent agents, represented as Java Bots in the UT2003 environment, is then set out. Java Bots can dynamically change their roles according to the situations (e.g., as a team member, as an individual helper, or as a team helper). A specific mechanism that interfaces Java Bots to UT2003 follows, with the aim of developing agents that can dynamically change their autonomy levels (e.g., 0%, 50%, or 100%). Finally, the paper concludes with a description of how the framework proposed in the study can be used to investigate various factors affecting perceived presence.

2. The UT2003 Game Environment

The framework breaks down into three components:

1. **The Java Applet –** this Applet is used to add new agents/bots into a game session and to communicate with them.
2. **The Startup File –** this program is used to initialize the server and a game to certain specifications. It can also be used as a log by agents, which can be viewed by the human player(s).
3. **UT2003** game environment – also referred to as the Unreal Tournament 2003 Game.

Figure 1 shows the architecture of the framework developed in this study. UT2003 is a game that was co-developed by Epic Games, Digital Extremes, and Atari. UT2003 provides a simulated 3-D world in the form of pre-designed, customizable 3-D maps. Each map is customizable through the use of the UT2003 map editor *UnrealED* that allows a level designer to custom build a terrain based environment containing elements such as doors, elevators, and water. A map also contains AI routes, referred to as navigation points, which an Agent can traverse when it needs to travel from one location to another. A finished map is then simulated through the UT2003 physics engine, which is capable of adjusting the gravity of a simulated environment.

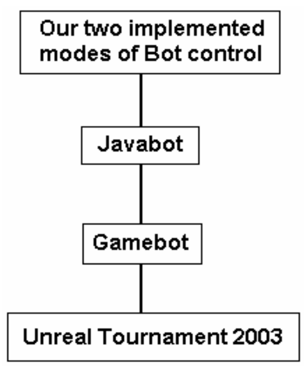

Figure 1 – The architecture of the framework

There are several ways in which one can play UT2003. The focus of our framework, however, is *Capture the Flag (CTF)*. In Capture the Flag, two teams (Red and Blue) attempt to outscore one another by capturing each others flag (see Figure 2). Each team has a *Home Base* in which their *Home Flag* resides. A team scores a point when they capture an enemy's flag, and return it to their home flag's spawn location.

The key to a team's success lies in maintaining the delicate balance between offense and defense. A team must be able to fortify its own strengths while exploiting the opposing team's weakness. In other words, a team's members must work together to ensure that their home flag remains safe while the opponent's flag is constantly under siege. In order for a team to work together, each member must have a line of clear communication to all other members. Through communication, a team is capable of maintaining an offensive and defensive balance via the division of offensive and defensive "roles" amongst its members. A typical successful team contains members that are willing to adjust offensively and defensively together according to the opponent's tactics.

Figure 2 – Example of *Capture The Flag* game

3. Intelligent Agents

In a UT2003 environment, Agents are referred to as Bots. Each Bot has been modified from it original UT2003 design to properly integrate into our framework. Each individually created Bot has its own intelligence, based on a uniquely designed team-oriented algorithm. In order to increase the probability of a team's success, each Bot is capable of serving as any of the following six unique offensive and defensive roles in our *Capture the Flag* game framework:

- ***CaptureEnemyFlag*** – A Bot will attempt to Capture and Enemy Flag to score a point for its team.
- ***DefendOurFlag*** **–** A Bot will travel to its Home Base Flag and wait until an approaching enemy is seen, at which point it will fight that enemy.
- ***DefendOurBase*** – A Bot will travel to the outskirts of its home base and wait until an approaching enemy is seen, at which point it will fight that enemy.
- ***SearchAndDestroy*** **–** A Bot will roam the map, increasing its Inventory and Health until an enemy is seen, at which point it will fight that enemy
- ***CoverMe*** – When a Bot issues this order, all available Bots will follow and protect the issuer until it dies.
- ***HoldThisPosition*** **–** When a Bot issues this order, all available Bots will travel to the position issued and wait until an approaching enemy is seen, at which point it will fight that enemy.

Crucial to the Bot AI framework is a Bot's ability to change roles. The core of a Bot's AI, the *Brain*, assesses each team member's role and surrounding environments, and makes a decision on what role would best serve the team. Additionally, Bot's are able to communicate clearly with one another – affecting Bot role assessment.

While in a specified role, a Bot will perform tasks related to that role. Under certain environmental situations, it is necessary for a Bot to reassess what actions it is performing in relation to its team members. Figure 3 demonstrates a code sample utilized in a Bot's brain.

```
if(DOF == 0 && CEF == 0)
{
    r = frand();
    if(r < 0.5) gotostate('DefendOurFlag','Begin');
    else gotostate('CaptureEnemyFlag','Begin');
}
else if(CEF > DOF || DOF == 0) gotostate('DefendOurFlag','Begin');
else if(DOF > CEF || CEF == 0) gotostate('CaptureEnemyFlag','Begin');
else if(DOB > DOF || DOB > CEF || SND > DOF || SND > CEF)
{
    r = frand();
    if(r < 0.5) gotostate('DefendOurFlag','Begin');
    else gotostate('CaptureEnemyFlag','Begin');
}
else
{
    r = frand();
    if(r < 0.5) gotostate('DefendOurBase','Begin');
    else gotostate('SearchNDestroy','Start');
}
```

Figure 3. Code sample from a Bot's Brain

The extent to which a Bot can independently reason is dependent upon a Bot's autonomy level. Before discussing Bot autonomy, let us first consider the manner in which Bots are interfaced to UT2003 and to a human user.

3.1. Interfacing to UT2003

Interfacing Bots to UT2003 is performed using the contributions made with the *GameBots* and *JavaBots* projects, as shown in Figure 4. *Gamebots* is a system which allows the UT2003 Bots in game are controlled using network sockets connected to clients. *JavaBots* provides a selection of Java packages that are designed for handling low level communication to the *GameBots* server. Essentially, the combination of *JavaBots* and *GameBots* allows any Java-based software to directly interact with UT2003. Additionally, *JavaBots* contains an API that allows a human user to connect and interact with UT2003 and its Bots therein.

The *JavaBots* API, which our development team has modified, provides two modes with which a human user can utilize for Bot control. In one mode, a human user has the ability to "command" a Bot to become a specific role (described in Section 3) by clicking an appropriate order related button. In the other mode, a user can control specific tasks (RunTo, Jump, Shoot, etc.) for a specific Bot. Selecting a task, providing appropriate data into an input box, and clicking the Send button will cause a Bot to immediately perform the selection. A Bot communicates with the human user in two ways. First, a Bot textually communicates with a human user through the UT2003 log window. Second, a Bot vocally communicates with a human user through the UT2003 game environment window. The purpose of a Bot communicating with a human user is to provide feedback on what a Bot is thinking or doing. Figure 4 shows the Java Client Window where the human user and Bots communicate each other.

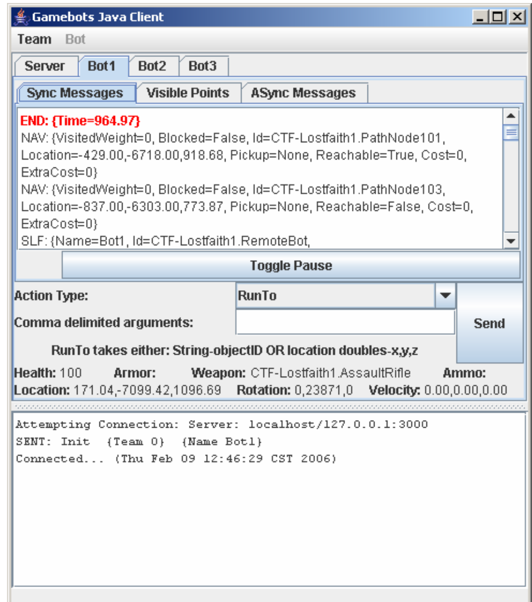

Figure 4. Java Client Communication Window

Finally, a *Capture the Flag* game is graphically animated in the UT2003 game environment window. By clicking the UT2003 window, a human user is able to cycle through each generated Bot's view. This allows a human user to visually see what a Bot is doing at a particular time.

3.2. Bot Autonomy

There are three levels of Bot autonomy, approximately labeled 0, 50, and 100 percent.

- **0%** - A Bot has little to no independent reasoning and must constantly rely on a human commander what role to take (*CaptureEnemyFlag* is the default).
- **50%** - A Bot has moderate independent reasoning, but must occasionally ask a human commander what role to take.
- **100%** - A Bot is fully independent, capable of reasoning what role to take on its own.

Regardless of a created Bot's autonomy level, a human commander can directly command a Bot by issuing a command through the JavaBots API interface.

4. Presence in technologically mediated environments

The framework developed in the study can be used to investigate individual's perceived presence and its affecting factors in technologically mediated environment, especially in the context of adaptable multi-agent system environments. The four interposed factors that may hinder or facilitate presence will be described: control, sensory, distraction, and realism.

4.1. Control Factors

The greater the level of control a user has, regarding their actions in an environment, the higher the level of presence [2]. The Control factors are determined by the following aspects:

- **Degree of control:** It is generally believed that the control over environment increases the presence level of a human user.

- **Immediacy of control:** The delay in the response of virtual characters decreases the presence level [5]. For example, a virtual character (or avatar) should be highly responsive to reciprocate a high level of presence to the human user.

- **Anticipation of control:** Human users will experience a greater sense of presence if they are able to anticipate the next action of a character (or avatar), regardless of whether it is under their personal control [5].

- **Mode of control:** If the user interacts with a system via effective modes, then the presence a user experiences increases. Additionally, the more modes of control available to a user, the higher level of presence.

- **Physical environmental control:** The degree to which an environment's physical objects can be manipulated by a user affects the degree of presence experienced by a user.

In our framework, the Interface and Bot autonomy directly affect Control. Our Interface provides a *Degree of Control* that provides a user with the necessary control to directly influence a Bot's role and many specific movements, yet a user cannot directly control all aspects of a Bot's movement. The *Immediacy of Control* is a mere fraction of a second. Any user-issued order is immediately performed by a Bot. Because a user can view a Bot during a *Capture the Flag* game, there is a higher degree of presence experienced through the *Anticipation of Control*. There are two effective *Modes of Control* (described in Section 3.1) a user can utilize for Bot Control. UT2003 maps provide *Physical Environmental Control* for pickup items, flags, water, and other Bots within a *Capture the Flag* game.

A Bot's autonomy level can have a direct affect on presence levels. The more autonomous a Bot is, the less it needs to be controlled. However, if a user is commanding a group of Bots, then a user's focus shifts from controlling one Bot, to controlling a team. Thus, the level of presence can vary based on a user's self-perceived role. Regardless, a user does have the ability to directly influence any single Bot.

4.2. Sensory Factors

Sensory information is the information humans receive through their sensors: eyes, ears, touch, etc. A greater degree of sensory information will lead to a higher level of presence [2]. Sensory is defined by the following aspects:

- **Sensory modality:** Different sensory modalities influence the degree of presence experienced by the human user. Visual sensory provides the greatest degree of presence, while other sensory channels provide lesser degrees.

- **Environmental richness:** The amount of information transmitted by system is proportionate to the level of presence experienced by a human user [1]. A vast array of environmental stimulations generates a greater sense of presence.

- **Multimodal sensory:** Senses stimulated in tandem by a system increased the presence experienced by a user [5]. Nam & Chung (2006) also showed that thermal feedback can facilitate the user's perceived presence in virtual environments [6].

- **Consistency of multimodal information:** The information received through all modalities should describe the same objective world [6]. If information from one modality gives a message that differs from that experienced through a different modality, a user's presence level will decrease [5].

- **Degree of movement perception:** If the user perceives self-movement through the system, to the extent that objects appear to move relative to the character, then the presence experienced by a user increases.

- **Active search:** The degree of environmental sensory control given to the user increases a user's presence level [1]. For instance, the extent to which users can adjust their viewpoint to change what they see, increases a user's presence.

In our framework, there exist visual, auditory, and haptic *sensory modalities*. A user receives visual feedback through all three interface windows (*JavaBots* API, UT2003 Log, UT2003 game), which can also be manipulated into different levels (e.g., desktop virtual environment Vs. Head-Mounted Display). A wide array of auditory feedback occurs through listening devices used by the user. Auditory feedback includes all auditory information relative to a viewed Bot within the UT2003 window (e.g. item pickup sound, gunfire, Bot footsteps, etc.), verbal communication relayed among Bots (e.g. "Cover Me", "I've got your back", etc.) as well as UT2003 in-game messages (e.g. "Flag has been captured", "Flag returned", "Flag dropped"). Using a joystick (e.g., Logitech force 3D pro joystick and a precision game controller) provides haptic feedback to a user while playing as well. These three modalities affect the *Environmental Richness* of UT2003. These modalities are used in tandem, increasing *Presence* through *multimodal sensory*. All information relayed through these modes describe the same objective world (mapped environment) in UT2003, upholding the *Consistency of Multimodal Information*. The *Degree of Movement Perception* is consistent through the use of *Third-Person Perspective* in relation to each individual Bot. A user can *Actively Search* through multiple onscreen characters by clicking the UT2003 game window.

4.3. Distraction Factors

Distraction factors are sensory impacts from the natural world, not the mediated environment [2]. Distraction Factors are defined by the following:

- **Isolation: S**ystems that isolate a user from their physical environment may increase the presence experienced by a user. For example, a head-mounted display that isolates users from the other disturbing factors may increase presence in the system in comparison to a standard two-dimensional, flat screen display.

- **Selective attention:** The observer's willingness or ability to focus on the stimuli as well as to ignore distractions that are external to the system directly affects the amount of presence experienced in that environment.

- **Interface awareness:** Unnatural interface devices interfere with the direct and effortless interpretation of (and interaction with) an environment [5]. Hence, an intuitive, natural interface will increase a user's presence.

4.4. Realism Factors

Realism factors represent the continuity and connectedness [in comparison to reality] of the user experience [2].

- **Graphical realism:** Graphical realism refers to the connectedness and continuity of the stimuli which is being experienced. *Presence* increases as a system's graphical realism more closely resembles photorealism. Examples of rendered graphical elements include: graphical content, texture, resolution, light sources, field of view (FOV), and dimensionality [7].

- **Consistency of information with the objective world:** The more consistent a system conveys information resembling real-world experiences, the more a user will experience presence.

- **Meaningfulness of experience:** Meaningfulness pertains to user motivation, task saliency, and previous experience. A more meaningful situation will increase user presence.

- **Separation anxiety/disorientation:** System users may experience disorientation or anxiety when returning from the system to the real world. The amount user disorientation is proportionate to the presence a user experiences in a system.

UT2003 directly affects Realism. The UT2003 game engine is solely responsible for generating graphical information in the UT2003 game window. Thus, UT2003 determines the *Graphical Realism* of our system. The UT2003's physics engine is responsible for the *Consistency of Information with the Objective World*. *Meaningfulness of Experience* and *Separation Anxiety/Disorientation* are determined by the *Presence* level of the individual user.

Conclusions

This paper described a research framework that can serve as a test-bed to presence research, while giving a basis for more flexible and systematic study on presence in technologically mediated environments. Therefore, the framework will allow presence (and even copresence) researchers to manipulate and investigate system configurations and parameters (e.g., dynamically changing agent roles, autonomy level, etc.), as well as psychological (e.g., personality, trust, etc.) and social factors (e.g., collective efficacy, etc.), which may hinder or facilitate the user's perceived presence in multi-agent contexts.

As it is still undergoing constant development, future work can be done in two directions. One is the implementation of more sophisticated development strategies for intelligent agents to be able to change their roles (e.g., as a team member, as an individual helper, or as a team helper) and autonomy levels (e.g., 0%, 50%, or 100%) according to the situations agents made sense. Another future work includes a series of empirical experiments to investigate how these technical factors and other variables (e.g., psychological and social variables, or interface types) affect the perceived presence of a human user(s) interacting with technologically mediated environments such as adaptable multi-agent systems.

Acknowledgements

This research was in part supported by the Air Force Office of Scientific Research (AFOSR) under Grant No. FA9550-05-1-0182. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Air Force Office of Scientific Research (AFOSR).

References

[1] T.B. Sheridan. Musings on telepresence and virtual presence. *Presence*. 1992, 1(1), 120-126.

[2] B.G. Witmer, M.J. Singer. Measuring presence in virtual environments: A presence questionnaire. *Presence: Teleoperators and Virtual Environments*. 1998, 7, 225-240.

[3] J. Steuer. Defining virtual reality: Dimensions determining telepresence. *Journal of Communication*. 1992, 42(4), 73-93.

[4] D.W. Schloerb. A quantitative measure of telepresence. *Presence.* 1995, 4(1), 64-80.

[5] R. Held, N.I. Durlach. Telepresence. *Presence: Teleoperators and Virtual Environments*. 1992, 1, 109-112.

[6] C.S. Nam, D. Chung. The influences of haptic thermal feedback on object identification and perceived presence. In *Proceedings of the 16th World Congress of the International Ergonomics Association (IEA)*. 2006. Maastricht, Netherlands: IEA Press (to appear).

[7] W.A. IJsselsteijn, H. de Ridder, J. Freeman, S.E. Avons. Effects of stereoscopic presentation, image motion, and screen size on subjective and objective corroborative measures of presence. *Presence*, 2000, 10, 298-311.