International Journal of Industrial Ergonomics 39 (2009) 192–201

Contents lists available at ScienceDirect

International Journal of Industrial Ergonomics

journal homepage: www.elsevier.com/locate/ergon

Evaluation of human–agent user interfaces in multi-agent systems

Chang S. Nam [a,*], Steve Johnson [a], Yueqing Li [a], Younho Seong [b]

[a] Department of Industrial Engineering, University of Arkansas, 4207 Bell Engineering Center, Fayetteville, AR 72701, USA
[b] Department of Industrial and Systems Engineering, North Carolina A&T State University, Greensboro, NC, USA

ARTICLE INFO

Article history:
Received 6 November 2007
Received in revised form 24 July 2008
Accepted 21 August 2008
Available online 25 October 2008

Keywords:
Human–agent user interface
Multi-agent system
Usability evaluation
Human factors

ABSTRACT

A multi-agent system (MAS) is a computational system in which two or more intelligent agents work together to perform a set of tasks. MAS's have had a significant impact in a wide range of applications. However, little attention has been paid to the design of human–agent user interfaces. Our research was conducted primarily to further explore interaction between humans and intelligent agents, as well as attempt to solidify foundational concepts for intelligent multi-agent system interface design. Based on the foundation of the commercialized PC game Unreal Tournament 2003, a framework was developed that utilized fundamental interface design principles as well as newly designed guidelines. Three user interfaces were designed for an experiment conducted with 25 participants. Their performances were evaluated via effectiveness, efficiency, satisfaction, and workload metrics. Through the testing of scenario tasks, an in-depth study was conducted on the collaboration between the human user and intelligent agents within a system. Finally, the paper summarized a set of design principles and guidelines developed through the study, which could easily be applicable to the design of future user interfaces for multi-agent systems.

Relevance to industry: A deeper understanding of the manner by which the appropriate information is provided to the user interacting with multi-agent systems in the correct form at the right time should have a broad impact on the future development of intelligent multi-agent system interfaces.

1. Introduction

A multi-agent system (MAS) is a computational system, in which human user(s) and two or more intelligent agents work together to perform a set of tasks (Brooks, 1991; Krupansky, 2003; Lesser, 1999). Intelligent agents are "software entities that carry out a set of operations on behalf of a user or another program with some degree of independence or autonomy" (Wickramasinghe et al., 2004, p. 603). An MAS can be utilized to reduce human cognitive workload in complex and time consuming tasks (Maes, 1994; Scholtz and Bahrami, 2003; Tambe et al., 1995). Generally, multi-agent systems are used for tasks that could be considered unreasonable or too costly for humans to perform and maintain (Laird and van Lent, 2001). MAS's have had a significant impact in a wide range of applications such as aircraft maintenance (Shehory et al., 1999), mission planning (Lenox et al., 2000; Pechoucek et al., 2002), traffic control (Hexmoor et al., 2006), information networks (Gerber and Klusch, 2002; Vlahavas et al., 2002), and education (Cobo et al., 2007). Such rapid growth of agent-technology applications has also generated a need for systematically designing interactions between the user(s) and agent teams. Because the agents perform tasks on behalf of the human user (e.g., information processing), an MAS should support the appropriate form of interaction between agents and the user (Bradshaw, 1997). For example, user interfaces should be designed in a way that allows the user interacting with the MAS to clearly know what an individual agent is currently doing, when intervention is necessary, and what information is needed to make effective interventions (Scholtz, 2002).

However, little attention has been paid to the design of human–agent user interfaces, including the way that the human user interacts with agents, as well as the manner by which the appropriate information is provided to the user in the correct form at the right time. Based on the foundation of the commercialized PC game Unreal Tournament 2003, a framework was developed that utilized both fundamental design principles and newly designed guidelines – such as the Layout Ranking Heuristic – in order to design and develop user-centered interfaces. These interfaces were used to enhance human collaboration with intelligent agents in a set of uniquely designed scenario tasks. Through the testing of these scenario tasks, an in-depth study was conducted on the collaboration between the human user and intelligent agents within the system. The primary objective of this study was to systematically

* Corresponding author. Tel.: +1 479 575 2563; fax: +1 479 575 8431.
E-mail addresses: cnam@uark.edu (C.S. Nam), sajohns@uark.edu (S. Johnson), yxl002@uark.edu (Y. Li), yseong@ncat.edu (Y. Seong).

0169-8141/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.ergon.2008.08.008

evaluate the overall quality of the user interfaces in order to obtain usability and performance data, as well as to develop design guidelines for the future development of intelligent multi-agent system interfaces.

This paper begins with an overview of the study's background – including a description of the research framework, an explanation the framework's Capture the Flag game, an overview of the framework's Intelligent Bots (IBots), and a discussion of the design and integration of user interfaces utilized to facilitate human interaction with intelligent multi-agents within the framework. Afterwards is a description of research methodologies and measurements. An analysis of results follows. Finally, the paper concludes with the study's conclusions and discussions.

2. Background

2.1. Information- and user-centered interface design

There are several issues that should be considered when designing user interfaces for multi-agent systems – most of which are information related. Information issues often result from not providing the user with the appropriate type and/or quality of information in the correct form at the right time to support domain tasks, as well as not recognizing that new user roles (e.g., supervisor, teammate, or bystander) necessitate new types of information (Malin and Schreckenghost, 1993; Scholtz, 2002). In order to utilize this information-centered user interface design approach and facilitate interface usability, it was essential in this study to determine types of information supported that would be deemed both necessary and appropriate by users for domain tasks. Additionally, an intelligent system's functionality must be "easily interpretable by a user in regards to the context of the system and its environment" (Suchman, 1987, p. 99).

When interacting with an intelligent MAS within a domain task, users and agents often change their roles dynamically in order to effectively respond to environmental stimuli and situations. In order to facilitate user interaction with an intelligent MAS, a user-centered interface design approach needs to be utilized to account for new user and agent roles – which would ultimately account for new types of information in the framework's already established information-centered design approach. Because the framework's user interface design approach was both information centric and user centric, it was crucial for the user interfaces to be able to update and display information in real-time. Likewise, any displayed information would need to be easily distinguishable by the user in order to avoid confusion. Because human decision tends to demand a large amount of information in order to make an appropriate analysis, the user interfaces needed to convey appropriate, high quality, user-centered information so that a human decision could be made quickly and efficiently.

2.2. Research framework

The framework breaks down into three components:

(1) *The Bot Connection Window* – this interface serves as the communication and control link between a human player and the UT2003 IBots.
(2) *Unreal Tournament Dedicated Server* – this program is used to initialize the UT2003 server for LAN play.
(3) *UT2003 Game Environment Window* – this interface renders and animates all graphical data in 3D using the UT2003 physics engine. UT2003 is synonymous with the Unreal Tournament 2003 game.

Fig. 1 shows the architecture of the framework developed in this study. UT2003 was selected as the platform of choice because UT2003 is open source, meaning that the entire base source code is available to developers.

2.2.1. Capture the Flag

There are several ways in which one can play UT2003, including Capture the Flag, Deathmatch, Bombing Run, and Last Man Standing. The focus of the framework, however, is Capture the Flag (CTF). In Capture the Flag, two teams (Red and Blue) attempt to outscore one another by capturing each other's flag (see Fig. 2). Each team has a Home Base in which their Home Flag resides. A team scores one point if and only if they capture the enemy's flag and return it to their home flag's location. The key to a team's success lies in maintaining the delicate balance between offensive and defensive strategies. A team must be able to fortify its own strengths while exploiting the opposing team's weaknesses. In other words, agent team members and the human operator(s) 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 clear line of 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 human user is able to cycle through each generated IBot's third person view, which allows the user to visually see what an IBot is doing at a particular time.

2.2.2. Intelligent Bots

In addition to unique intelligence, each IBot has an autonomy level. An autonomous IBot can be defined as "a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future" (Franklin and Graesser, 1996, p. 4). The extent to which an IBot can independently reason and act is dependent upon an IBot's autonomy level. In critical situations, less autonomous IBots need human input in order to assess what actions they should perform. In this study, two levels of IBot autonomy were implemented labeled medium level (50%) and full level (100%). An IBot's autonomy is only controlled by a human user.

- *Medium level (50%)* – an IBot has moderate independent reasoning, but must occasionally ask a human player what to do in critical situations.
- *Full level (100%)* – an IBot is fully independent, capable of reasoning what role to take on its own as well as what actions to take in critical situations.

2.3. User interface design goals for IMAS

In order to realize the framework's information- and user-centered design approaches, a set of user interface design goals were established – a fundamental principle in user interface design methodology. Interface design goals included:

(1) Display in one window all of the necessary and appropriate information for a team of four IBot agents and one human player in order to facilitate interaction.
(2) Layout the information in a user friendly, easy to read fashion.
(3) Provide essential, non-overwhelming information to the user to assist in efficient decision making.
(4) Maximize usability by facilitating rapid user interaction with a team of IBot agents.
(5) Provide a user with necessary and appropriate interface information and interactions to facilitate the dynamic role changing that will occur by both the user and the IBot agent(s).
(6) Provide a user with interaction feedback.