Advanced Engineering Informatics 33 (2017) 410–424

Contents lists available at ScienceDirect

Advanced Engineering Informatics

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

Full length article

Are you a human or a humanoid: Predictive user modelling through behavioural analysis of online gameplay data ☆

Chen Gao [a], Kaiqi Jin [a], Haifeng Shen [a,*], Muhammed Ali Babar [b]

[a] School of Computer Science, Engineering and Mathematics, Flinders University, Adelaide, Australia
[b] School of Computer Science, University of Adelaide, Australia

ARTICLE INFO

Article history:
Received 27 June 2016
Received in revised form 9 January 2017
Accepted 13 January 2017
Available online 7 February 2017

Keywords:
Intelligent agent
Human modelling
MOG
Latency
Jitter
QoE
Humanoid bot
Behavioural analysis
Gameplay data
Predictive user modelling
Regression analysis
Bayesian network

ABSTRACT

Intelligent agents are widely used in robotics, gaming and simulation. A key issue is modelling human behaviours so that intelligent agents can use a human's behavioural model to imitate them and predict their next moves. In this article, we use Internet-based multiplayer online gaming (MOG) as a case study to present our approach to predictive user modelling through behavioural analysis of online gameplay data. As latency is an inherited bottleneck of the Internet and is likely to remain so into a foreseeable future, a lot of efforts have been made to address the resulting issues. Most of the existing latency handling techniques are based on the assumption that latency is within an acceptable threshold so that they can alleviate or even completely hide its negative impact on players' quality of experience (QoE) that directly determines consumers' satisfaction of the provided MOG services. While this assumption is mostly valid, it is worth noting that a player's Internet connection latency always fluctuates (known as jitter), possibly to the extent of exceeding a MOG's designated threshold in which case none of the techniques can handle properly but disconnecting the player from the gameplay session. Forcing a player to quit prematurely simply due to a spike of unusual high latency has a significant negative impact both on the gameplay's fairness and on the player's QoE. To improve customer satisfaction of a MOG service, we propose a more tolerant approach by temporarily substituting a player with a humanoid bot in the event of latency hikes so that the player always remains in the gameplay session. The challenge in this approach is to create a personalised humanoid bot that can imitate the playing pattern of the individual human player being substituted. Our solution is to first extract key variables that have impact on the human player's decision-makings through behavioural analysis of the player's historical gameplay data, then model the relationships among these variables, and finally creates the player's humanoid bot with the model. In this paper, we use a multiplayer online pong game as a case study to explain behavioural variables, modelling techniques, processes, outcomes, and performance studies.

1. Introduction

There has been a long history of computer agents imitating humans ever since the inception of artificial intelligence (AI) [1], with IBM's Deep Blue Chess Computer [2] being one of the most-cited success stories. The recent AlphaGo programme developed by Google DeepMind extended the glory by beating a professional human Go player without handicaps [3], prompting humans to be concerned with the AI takeover. In these examples, the main objective of AI is to create a super-intelligent agent that can outperform humans.

However, another type of application of AI is equally important if not more, which is to create an intelligent agent that can imitate humans, such as assistive or factory robots. A key to this issue is modelling human behaviours so that intelligent agents can use a human's behavioural model to imitate them and predict their next moves. Specifically in the domain of engineering informatics, work has been done on lifelike assistive robotics [4], intelligent machine agents for adaptive control optimisation of manufacturing processes [5], and modelling and simulation of pedestrian crowd behaviour [6]. Nowadays intelligent agents are widely used in robotics, gaming and simulation.

☆ This article is a significant extension to the paper published at the 20th IEEE International Conference on Computer Supported Cooperative Work in Design (CSCWD 2016).

* Corresponding author.
E-mail addresses: gao0141@uni.flinders.edu.au (C. Gao), jin0098@uni.flinders.edu.au (K. Jin), haifeng.shen@flinders.edu.au (H. Shen), ali.babar@adelaide.edu.au (M.A. Babar).

http://dx.doi.org/10.1016/j.aei.2017.01.004
1474-0346/

In this article, we use Internet-based multiplayer online gaming (MOG) as a case study to present our approach to predictive user modelling through behavioural analysis of online gameplay data. We start with the background of MOG and the main issue we want to tackle. Over the past decade, the MOG industry has been developed in a rapid pace and attracted enormous attention. A salient benefit of MOG is that one can easily find partners on the Internet to play with/against anytime and anywhere [7]. MOGs of various genres are published every month for people to collaboratively play against powerful monsters or compete with each other without considering geographical distance, such as First Person Shooter (FPS), Role Playing Game (RPG), and Real Time Strategy (RTS).

As these MOGs are delivered over the Internet, latency, which is measured as Round Trip Time (RTT) between a MOG client and the MOG server in a client/server architecture or between two MOG clients in a peer-to-peer architecture, presents a major hinderance on gameplay quality of experience (QoE) [8] that directly determines consumers' satisfaction of the provided MOG services. Responsiveness and consistency are two important QoE factors directly influenced by latency [9]. Consistency refers to that each node on the network has the same game state, while responsiveness refers to the delay for an update to register throughout the network [10]. However, responsiveness and consistency are often two contradictory goals as meeting one may compromise the other. A classic example of inconsistency is seeing a dead man shooting in a game world. Although the game is responsive as the local node instantly determines a man has been shot dead, this message has not reached other nodes, where the man is still perceived alive. An opposite example is observing a lag between firing a weapon and perceiving the visual effect of this action as the message needs to reach the game server, which confirms the action before allowing nodes to update the visual effect. Although consistency has been maintained as all nodes observe the same lag, all the players would feel the game is irresponsive and may decide to terminate the gameplay.

All MOGs have a stringent responsiveness requirement, although different games can tolerate different thresholds, for example, 100 ms for FPS, 500 ms for RPG, and 1000 ms RTS [11]. Highly interactive MOGs such as action games, shooter games, platform games and sports games have a very low responsiveness requirement [12] and consequently give greater weight to responsiveness. In contrast, MOGs that do not require high responsiveness such as RPG and RTS tend to put greater emphasis on consistency. Generally speaking, existing latency handling techniques are mainly focused on either consistency or responsiveness. For example, time manipulation based techniques, such as lockstep [13], local lag [14], time warp [15], and progressive slowdown [16], aim at maintaining consistency at the cost of responsiveness, whereas prediction based solutions such as dead reckoning [17] and speculation [18] have a totally opposite objective. Nevertheless, these techniques are often combined, together with techniques aiming to alleviate communication latency such as gossiping [19] and optimistic obsolescence-based synchronisation [20], in order to ensure both responsiveness and consistency (whichever has the higher priority) are acceptable by players to warrant a reasonable gameplay QoE.

Common to these solutions is the assumption that latency is within an acceptable threshold so that these techniques can alleviate or even completely hide its negative impact on players' QoE. While this assumption is mostly valid, it is a fact that the latency of a player's Internet connection always fluctuates (known as jitter), possibly to the extent of exceeding a MOG's designated threshold in which case none of the techniques can handle properly but disconnecting the player in the middle the gameplay session. For example, when a player's RTT exceeds 130 ms for a FPS game, they will be forced to leave the game, which makes sense only if their latency is constantly well above this threshold. In contrast, if the latency is mostly below the threshold, a spike of unusual high latency forcing the player to quit prematurely would negatively impact the fairness of the game as well as the gameplay QoE. To improve customer satisfaction of a MOG service, we propose a more tolerant approach by temporarily substituting a player with a humanoid bot in the event of latency hikes so that the player always remains in the gameplay session [21].

A major challenge in this approach is to make each humanoid bot imitate the playing pattern of the individual human player it substitutes. Our solution is to first extract key variables that have impacts on human players' decision-makings through behavioural analysis of a particular player's gameplay data, then derive a predictive user model that captures the relationships among these variables, and finally creates a personalised humanoid bot for the player with the model. We use a multiplayer online pong game as a case study to showcase the proposed approach and more importantly explain modelling techniques, processes, outcomes, and performance studies.

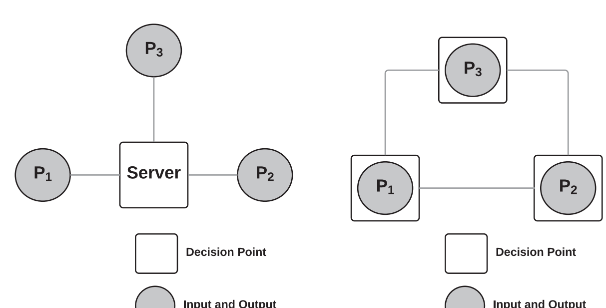
The rest of the paper is organised as follows. Section 2 discusses some related work on MOGs and latency handling techniques. Section 3 outlines the concept of predictive user modelling through behavioural analysis in the context of a pong game. Sections 4 and 5 present two modelling methods using regression analysis and Bayesian network techniques respectively. Section 6 finally concludes the paper with a summary of major contributions and future works.

2. Related work

2.1. Network architecture

Two types of network architectures are commonly used for MOGs: client/server and peer-to-peer. The most widely used architecture is the client/server architecture where clients do not make decisions on game states or communicate with each other directly. As depicted by Fig. 1(a) [22], a player node is mainly a dumb input/output client that accepts input from the player, transmits the input to the game server, receives the new game state from the server and renders the new state to the player. As a single server may become a bottleneck, which inherently disadvantages player nodes that are geographically further away from the server, most of today's MOGs deploy multiple mirrored or clustered game servers [23].

Most of the early networked games and military simulation systems were based on the peer-to-peer architecture, which in contrast to the client/server architecture does not rely on a central game server. As illustrated by Fig. 1(b) [22], every player node maintains a full game state and makes independent decisions on



(a) The client/server architecture (b) The peer-to-peer architecture

Fig. 1. Network architecture.