Capturing and Storing Profile Information for Gamers Playing Multiplayer Online Games

Tomas Hildebrandt Sonja Bergsträßer Christoph Rensing Ralf Steinmetz
Multimedia Communications Lab - KOM
TU Darmstadt
Darmstadt, Germany

Abstract—**This paper presents methods for capturing and storing gaming related context information into online gamer created profiles. Captured information may not be manipulated while being processed and stored within user created profiles and therefore is reliable. We propose a concept of client side capturing of profile information within the working environment of a gamer playing online games and a profile server storing generic community-extendable gaming profile data models that can be individualized. This addresses the creative power of gaming communities and faces the challenge to define machine processable data structures that can be updated with context information. A gamer created profile, enriched with detailed usage and reliable context information, improves the expressiveness and keeps it more up-to-date.**

Networked Gaming, Online Games, Community, Gaming Profiles, Profile Information Capturing, Context

I. Introduction

The current trend for self-representation within a certain community is to maintain detailed personal profiles. A profile contains information representing a user and his personal social network. Other users, interested in a user or his provided information, activities and contacts should be able to find him, read his profile and finally contact him.

Social aspects are important in the area of *multiplayer online gaming* [1] and with emerging social networks in mind, a gamer is interested in maintaining an *up-to-date profile*. Such a profile includes information about *all* his played games, in-game characters or identities and his personal social network. In addition and as a good starting point for gaming activities, he wants to add *communication* and *collaboration awareness* and used web resources or links to tools. In contrast to social networks from other information domains most of this information exits *digitally* and can be *automatically enriched* or *updated* and therefore is *reliable*. For example, if and how long a gamer plays a certain game can be easily tracked by observing the running processes at the user's operating system. Some games or community platforms started creating sites providing information about in-game achievements, progress or equipment. This information is reliable if the source is trustworthy. Putting all this information together, it leads to a gaming profile with more expressiveness and actuality. The challenge is to design a global gaming profile data model that aggregates existing profile information and to implement a concept for reliable enrichment of automatically captured information.

We analyze the gaming environment of gamers playing online games to find out, what information details ought to be part of a gaming profile and design a generic gaming profile data model that allows gamers to create a profile of their own gaming activities which can be shared within a community but remains machine processable to be extended by automatically captured information (section 2). We develop and implement a concept to capture and transfer generated profile information including real-time awareness to the centrally stored gamer profile (section 3). Finally we summarize the paper and give an outlook on future work (section 4).

II. Gaming Profile Information

A gaming profile consists of information that represents a gamer, including his played games, used tools and resources, his virtual game identities and his personal social network. It should be possible to add all this information to our global gaming profile. Additionally the derived generic profile data model of the gaming profile has to consider upcoming game releases and even completely new game genres and user activities. The following **profile information types** have to be considered:

User Information: User created information about preferences in gaming, games, characters, applications and communities.

Game information: Metadata about a game (name, publisher, etc.) including gameplay edge conditions (versus mode, group size, etc.).

Character Information: User created information about the virtual character (e.g. character class, in-game achievements, equipment, possible playable roles).

Application Information: Metadata about the application (e.g. name, application category like messenger or gaming voice system) and account information.

Context information: Captured Information about application and game usage and context features (e.g. awareness state) within the gaming environment.

Virtual context information: Captured in-game information about game accounts and characters (e.g. in-game location information, gameplay state, played role).

978-1-4244-5605-5/09/$26.00 ©2009 IEEE

Relationship information: Captured or user created information about friends, opponents and associations to groups (e.g. teams, groups of interest)

To respect the gamer's privacy, capturing and storing information should remain completely visible to the user. Additionally, profile information ought not contain personal data about a real user behind a gamers identity.

Profile information types can be allocated to different information layers, **game characters** builds the bottom layer then the **games and applications** layer and above the **gamer profile** layer. The **personal social network** is a layer on top of the gamer's profile and consists of lists of identities and links to groups. Lists of identities contain acquaintances or friends and groups are favorite teams and common interest groups. The personal social network layer links the gamer profiles to the overall **gaming social network**.

The profile information of a gamer is linked to a multitude of identities including the gamer himself. The **Gamer Identity** (*GamerId*) is the main root identity, where all information of a gamer profile is linked to. Its the identitiy of the real user. Additionally gamers often use a so-called *nickname* to represent themselves within the community. The **Net Identities** (*NetId*) includes identities used in the internet to identify a user against a game or an application, like a messenger account. **In-Game Identities** (*InGameId*) are the identities of game characters represented by a named avatar in a virtual world.

For a structured representation of personal gaming activities these three identity levels have to be considered. This leads to a **hierarchical profile data model** with **profile information nodes**. As stated before, reliable and up-to-date game and character information is crucial in a gaming profile. A static data model can not be customized by the user and a completely generic structure is not processable anymore, especially when every user creates his own structure. So we define a base structure for game, application and character profile information nodes with **mandatory information** parts. In addition, the user can add **user information** nodes for own purposes. To simplify automatically enrichment and processing of user information nodes we have developed a template concept. Every user information node becomes a template for the respective game, application or character. When another user wants to add a user information node to a profile, we recommend the existing templates to encourage reuse. All nodes are technically identified by *unique ids*. This allows manipulating nodes without knowledge of the full profile tree.

III. Information Capturing Architecture

By analyzing the gaming activities in the **gaming environment**, we have identified the gaming device as the main spots where profile information arises. The different applications are the local game client on PC or consoles, internet browsers like Firefox, messenger tools and gaming voice chat systems, game related collaboration tools, local documentation tools and connected gaming hardware devices. For capturing profile information within the gaming environment we propose a **software sensor concept** adapted from Lokaiczyk et al.[2]. Each running game or application can be detected by analyzing the Windows task manager list. Usage information can be captured by implementing sensors matching the process name of a certain game or application. This basic **process sensor** can also be instantiated and configured by the user. If the process name changes, the sensor should also trigger and detect the usage. A simple list of possible process names for a usage sensor solves the problem. If a process sensor identifies a game or application providing an API the respective **context sensor** is instantiated. Context sensors can capture, for example, instant messenger awareness by detecting the user's online state. **Virtual context sensors** work as adapters to game APIs and can capture virtual context information (e.g. in-game location, played role) as proposed by Bergsträßer et al. [3]. A bigger challenge is posed by internet applications accessed with a browser. A process sensor detecting the running browser is not very helpful by indentifying the accessed community page or wiki. Capturing usage information means analyzing all accesses to web pages with the browser. This notably violates the user's privacy because information is captured that does not belong to the gaming environment. Thus the user can configure a whitelist, mapping URLs to web applications or sites.

Our profile data model distinguished between game, application and character nodes. Each sensor captures information for a single node updating or extending it by reliable captured information. Because we can access each node by its unique id we can directly bind a sensor to a node. Virtual context sensors capturing information are creating, updating or extending several information nodes, which are below the character information node in the node hierarchy.

IV. Summery and Further Work

Nowadays social networks presenting personal user profiles are part of our digital life. Automatically captured profile information is reliable and updates or extends user created profiles for more expressiveness and actuality. In the digital world of networked gaming this information can be captured where it arises using our concept of software sensors. We therefore have defined and analyzed the gaming environment of gamers playing online games from a technical perspective and identified all profile information types of a gaming profile data model. Our presented and implemented architecture captures information with the client side application *jumpad.* This information is mapped directly to the profile data model on the server application *gamerlist* and updates or extends gamer profiles. We have also addressed the creative power of the gaming community. External and existing profiles can be added and sensors can be defined by the gamer himself. Additionally, gamers can individualize their profiles with user created information. Future work includes the evaluation of our template approach and search approaches.

[1] Ducheneaut, N., Yee, N., Nickell, E. and Moore, R.J., 2006: "Alone together?": exploring the social dynamics of massively multiplayer online games. In Proceedings of SIGCHI, April 2006, Montréal, Canada

[2] Lokaiczyk, R., Faatz, A., Beckhaus, A. and Goertz, M., 2007: Enhancing Just-in-Time E-Learning Through Machine Learning on Desktop Context Sensors. In: Modeling and Using Context, August 2007

[3] Bergsträßer, S., Hildebrandt, T., Rensing, C. and Steinmetz, R. 2009: Virtual context based services for multiplayer online games to facilitate community participation. In: Multimedia Tools and Applications, May 2009