Sonja Bergsträßer, Tomas Hildebrandt, Christoph Rensing, Ralf Steinmetz:
A Middleware for the Controlled Information Exchange between Online Games and Internet Applications. In: Klaus David, Kurt Geihs: 16. ITG/GI - Fachtagung Kommunikation in Verteilten Systemen 2009 - KiVS 2009, Kassel, Germany, Springer , March 2009. ISBN 978-3-540-92665-8.
http://www.springerlink.com/content/78t28j1672k72021/.

A Middleware for the Controlled Information Exchange between Online Games and Internet Applications

Sonja Bergsträßer[1], Tomas Hildebrandt[1], Christoph Rensing[1], Ralf Steinmetz[1]

[1]KOM – Multimedia Communications Lab, TU-Darmstadt {sonja.bergstraesser, tomas.hildebrandt, christoph.rensing, ralf.steinmetz}@KOM.tu-darmstadt.de

Abstract. Multiplayer Online Games (MOGs) are a thriving market leading to a multiplicity of game related internet applications. Enabling information exchange between games and these applications is essential, but still an unsolved challenge. Our Virtual Context Based Service (VCBS) middleware enables such an information exchange. The VCBS middleware is an interface between games and other internet applications, which also supports community activities. In this paper we describe the architecture of the VCBS middleware and its components, and introduce our concept for generic MOG interfaces.

1. Introduction

Today most internet applications are client-server applications (e.g. websites, networked virtual environments, voice-chat tools or shared calendars). Especially networked virtual environments like Multiplayer Online Games (MOGs) are isolated from other internet applications. Nevertheless various internet applications related to MOGs develop around these games, e.g. community portals, knowledge bases, communication and collaboration tools. We call these internet applications and related services the environment of MOGs. Normally there are no interconnections between the different servers and between different client applications and the data contained in one application can not be used in another one.

While there are first approaches for the utilization of game related information, yet no generic concept for information exchange between MOGs and other internet applications exists. We have developed such a concept which in addition supports community activities [1]. Based on this concept we introduce a middleware solution, the Virtual Context Based Service (VCBS) architecture, to enable data exchange between game clients and servers and other internet applications and to provide means for community participation. In addition our middleware includes a data flow control mechanism to avoid abuse of game related information. Cheat prevention is a serious issue in gaming research and discussed in various papers (e.g. [2, 3]).

We give an overview on different attempts to enable information access or user participation in the MOG area in section 2. Section 3 introduces virtual context based services and describes the VCBS middleware in detail. In section 4 we show how our architecture can be utilized. Finally we summarize this article in section 5.

2. Related Work

Hardly any general concept exists, about how information exchange between a game and its environment can be realized. Also the involvement of game communities is quite limited in most cases. For both scenarios only isolated applications and specialized approaches can be found. In the following we give a short overview of the current situation and introduce some of the existing approaches.

The users of MOGs form communities which actively participate in the evolution of games and shape them through the creation of large data collections, guidelines, and even custom tools and game add-ons. The commitment of the gamers to their games is generally very strong and the exchange and communication between gamers are essential elements. The request of game communities for creative participation is increasing and possibilities to adapt games and the gaming experience to the user will be a crucial factor for the success of a MOG in the future. Today we are already aware of the fact that fast growth of user count and binding a critical mass of users to a game is not only important for its development but decides on its success or failure [4]. The support of community development within the game [5] and in the game environment improves the binding of users to the game and thus is one success factor. Especially in the Massively Multiplayer Online Role-Playing Game (MMORPG) market, which is highly competitive, only MMORPGs with strong communities can be successful.

In some genres like First Person Shooters (FPS) the game industry has begun to open itself to the game community and some games already include editors to modify the game itself. But especially MMORPGs, which are based on player interaction and live through their communities, shield themselves from the outside world. One approach to provide selected ingame information is the developer driven game data presentation. Special internet portals where e.g. character information can be retrieved are operated by different game developers or publishers (e.g. WoW Amory [6] or ET:QW Stats [7]). A different approach is the usage of launch platforms coupled with community applications. This approach mainly focuses on awareness aspects and teaming and emphasizes communication applications (e.g. ESL Wire [8] or Steam Community [9]). Additionally third parties begin to utilize the opportunities given (e.g. World of Warcraft is providing an API which allows the retrieval of stored ingame information) and present available ingame information on web portals or community platforms (e.g. buffed [10] or xchar [11]).

The main drawbacks of current approaches beside the lack of a generic solution are coupling to one information source, specialization of applications, and no time synchronous or real-time data usage. The approaches are each strongly coupled to one kind of information source only, although only very few and limited information is available at all. The applications are very specialized: developer driven data representation portals only present data (no interaction is possible), launch platforms are mainly server browsers combined with buddy lists and web portals include community features enriched with limited game information. The time synchronous usage of data is not possible and thus no data exchange is feasible while a gamer is playing a game. In addition there are no concepts for the prevention of cheating while ingame information is exchanged with the game environment. In order to prevent

cheating most of the games simply do not allow any outside connection to the outside or retrieval of ingame information from the outside at all.

3. Virtual Context Based Services

The goal of Virtual Context Based Services (VCBS) is to provide services to the user regarding his current situation in a virtual environment [1]. The VCBS enable information exchange between a virtual environment and other applications. In addition the VCBS include a control mechanism for the information flow and thus complies with the given requirements. In the following we subsume the VCBS concept, shortly introduce our generic game interfaces and describe the architecture of the VCBS middleware and its components.

3.1 Virtual Parameters and Virtual Context

The virtual situation of a gamer's character (or avatar) can be described using different parameters, e.g. location in the virtual world, current action or character properties. These virtual parameters specify the virtual context of a user and his character. We have defined an XML based virtual parameter description language which is valid for MOGs in general. Thus generic game interfaces can be described and extended for specific genres and if needed also for specific games. This allows us to define an abstract interface for MOGs and include special needs of different genres and games. The virtual parameters are the base for the definition of an interface between games and other internet applications. We have defined four parameter sets (general, location, character and status) specifying the different categories of virtual parameters. The general set provides information about the player (e.g. player name), the game time (e.g. session time) and the server (e.g. server name), the other three sets about the virtual character and his situation in the virtual world. The location set provides position and area information (e.g. area name), the character set includes general information about the virtual character (e.g. skills) and the status set is about the current state of the character (e.g. health) and includes grouping information.

3.2 VCBS Middleware Architecture

The VCBS middleware consists of a **VCBS Client** on the client system, a **VCBS Registry** and **VCBS Servers** (see Figure 1). It also defines an interface for external services. The different services (e.g. a web platform providing gamer statistics, a wiki including situation related guidelines or a voice tool for group communication) are not part of the VCBS architecture and can be provided independently. This gives anybody the opportunity to provide services (e.g. game developers, the game community or third parties). Services are coupled with the game through the VCBS middleware. Hence they can utilize ingame information, react on ingame events and provide data regarding the current situation of a gamer.

The **VCBS Registry** manages all parts of the VCBS architecture, the registered services and user accounts. Before a service can be used with the VCBS architecture it has to be registered with the VCBS Registry (*service registration* (1)). The

registration data includes a service description which is stored in the Registry. The Registry checks during service registration, if a service is authorized to get the requested client information based on the rule set defined by the game (to check if the requested information is approved for the situations the service is designed for). It distributes the users to the VCBS Servers (to ensure the scalability of the system) and maintains the servers using *status updates* (2) (e.g. "new appointed gamer").

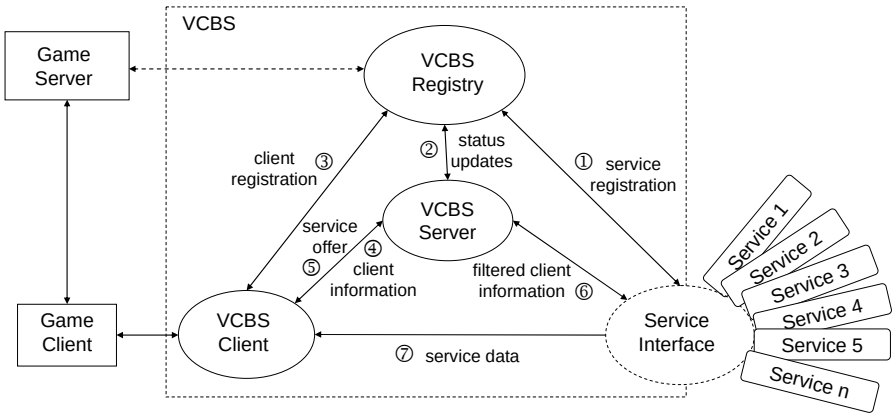

Figure 1. VCBS middleware: architecture overview and information flow

The **VCBS Client** is located at the gamer's system (client system). It handles the registration of the user (*client registration* (3)) and interacts with the user. The VCBS Client receives the virtual parameters from the game and integrates outgame service data into the game. The game interface of the VCBS Clients use plug-ins to handle different game APIs. When a user is online in a virtual world of a supported game, the VCBS Client regularly sends the game client information containing the virtual parameters (*client information* (4)) to the assigned VCBS Server.

The **VCBS Server** receives the game client information updates from the VCBS Client and offers available and suited services to the user (*service offer* (5)) through the VCBS Client. The Servers are responsible for the processing of the game client information. They decide which services are offered to the gamer and which information is forwarded to the external services. The matching is done based on the virtual situation of the user and two options for matching virtual situations to services are included in the VCBS middleware: parameter and event-based matching. Using parameter matching the VCBS Server compares the game client information with the service description. With event-based matching the VCBS Client reacts on events indicating changes of the user's context in the game. There is also the possibility to use other matching schemes. If a matching scheme triggers, the VCBS Server verifies the corresponding service, checks if the parameters are approved regarding the game based rule set and offers it to the gamer. Thus a control mechanism based on the virtual situation of a user is integrated into the information flow. If a gamer chooses to use a offered service, the VCBS Server forwards the relevant parameters to the corresponding service (*filtered client information* (6)). Then the service can process the client information, utilize the information or send his resulting data (*service data*

(7)) to the VCBS Client. The data is then integrated into the game by the VCBS Client. If the situation of the user changes and the service is not matching the situation anymore the VCBS Server stops sending the *filtered client information*.

4. USAGE SCENARIOS

There are multiple usage scenarios for the VCBS middleware which can be mainly divided into two categories. In the first category are internet applications using status information about the gamer outside of the game (e.g. at a web portal like gamerlist.net where a gamer wants to show information about himself to the community). The second category includes services which provide ingame user support.

We are currently developing "**gamerlist.net**" a community platform where gamers present themselves, their games, their relationships to other gamers and their community activities. It is used to increase the visibility of a gamer and his activities for the game community. This information can be used to support grouping, provide statistics, enable the comparison of gamers with others and so on. The platform utilizes the VCBS middleware in order to access ingame information, e.g. to show which game a user is currently playing, display his game statistics or (if possible and allowed by the game) his current ingame location. The ingame information is obtained by a service, which is part of gamerlist.net. This ensures that only information which is allowed to be shown is displayed on the platform and that the information access is standardized.

The **Service User Interface** (SUI) is an extension of the UI of a game. To include a service into a game an ingame representation of the service must be provided. With this ingame service representation, the user can interact and access the service data. An ingame representation of a service can be realized with the SUI. An example service we are using with the SUI is the “Fisherman's Friend” service. It supports a user when he goes fishing in a virtual environment. When the user’s character is standing at a virtual lake or river and takes a fishing rod in his hands the “Fisherman's Friend” service is offered to him. The service offer is based on the virtual context parameters, the coordinates and the used equipment in this example. The service data can include the fishes that can be caught there, the probability of catching the different species, whether the difficulty of the waters is suited for the character skill or which other fishing sites might be interesting to go to. To obtain this information (which is normally not provided directly by the game itself) the client information is collected by an external service while a character is fishing in the virtual world. The information is processed, and with more people using the service the obtained information is improved. The data from the service is embedded into the game by the VCBS middleware.

5. SUMMARY

The information exchange and service invocation between different virtual environments and internet applications is an important issue, especially concerning

gaming and game worlds. The evolution of MOGs in that direction is currently just at the beginning. Isolated approaches exist but they do not satisfy the needs. All approaches presented in the related work section are only first and non-systematic steps to open games to other applications or community participation.

This creates needs for a generalization of interfaces and communication mechanisms, which satisfies the interests of the community for flexible influence opportunities and complies with the requirements of the game industry for a protection of the game against forbidden modifications and the ensuring of fairness. It is important to consider the unique features of the different games or genres. But it is also important to find similarities and include the wide variety of games and gamers.

With the Virtual Contact Based Services and the VCBS middleware we offer the first generic connection of MOGs with internet applications. By offering a loose coupling our solution can be used to support all networked gaming scenarios. In conclusion the VCBS satisfy the identified requirements by enabling information exchange between games and internet applications and opening games to community participation while preventing abuse of game related information. We have already tested our VCBS concept with games which provide an API where we can obtain virtual parameters or ingame events. With our solution we want to provide game developers a framework so that they can open their games for the participation of the game community and use the high creative potential and developing power.

6. References

1. Bergsträßer S., Hildebrandt T., Lehmann L., Rensing C., Steinmetz R. Virtual Context Based Services for Support of Interaction in Virtual Worlds. In: Grenville Armitage: Netgames 2007, 6th Annual Workshop on Network and Systems Support for Games, p. 111--116, Melbourne, Australia, September 2007
2. Chambers, C., Feng, W., Feng, W., and Saha, D. 2005. Mitigating information exposure to cheaters in real-time strategy games. In Proceedings of the international Workshop on Network and Operating Systems Support For Digital Audio and Video (NOSSDAV '05), Stevenson, Washington, USA, June 13 - 14, 2005
3. Li, K., Ding, S., McCreary, D., and Webb, S. 2004. Analysis of state exposure control to prevent cheating in online games. In Proceedings of the 14th international Workshop on Network and Operating Systems Support For Digital Audio and Video (NOSSDAV '04), Cork, Ireland, June 16 - 18, 2004
4. Strain, Jeff: How to create a successful MMO, Games Convention Developers Conference (GCDC), Leipzig, 2007, http://eu.guildwars.com/press/article/jeffgc2007/
5. Ducheneaut, N., Yee, N., Nickell, E., and Moore, R. J. 2007. The life and death of online gaming communities: a look at guilds in world of warcraft. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, San Jose, California, USA, April 28 - May 03, 2007
6. The World of Warcraft Armory, http://eu.wowarmory.com/, last accessed 12.09.08
7. Enemy Territory: Quake Wars Stats, http://stats.enemyterritory.com/, last accessed 12.09.08
8. ESL: ESL Wire, http://www.esl.eu/de/wire, last accessed 10.09.08
9. Steam Community, https://steamcommunity.com/, last accessed 10.09.08
10. buffed.de: Das Portal für Onlinespiele, http://www.buffed.de/, last accessed 12.09.08
11. xchar: WoW meets real life, http://www.xchar.de/, last accessed 10.09.08