Virtual Context Based Services for Multiplayer Online Games to Facilitate Community Participation

Sonja Bergsträßer * Tomas Hildebrandt * Christoph Rensing * Ralf Steinmetz

KOM - Multimedia Communications Lab, TU Darmstadt, Merckstr. 25, 64283 Darmstadt, Germany

Phone: +49 6151 16-6150

Fax: +49 6151 16-6152

{sonja.bergstraesser, tomas.hildebrandt, christoph.rensing, ralf.steinmetz}@KOM.tu-darmstadt.de

www.kom.tu-darmstadt.de

Abstract In the future of Massively Multiuser Online Gaming new improvements and innovations are needed. With demands of gamers for support and involvement, increasing new approaches are required in order to enhance community participation and increase the gaming experience. Using the concept of Virtual Context Based Services we have developed a novel approach to solve these challenges. Our concept enables information exchange game and genre independent by using the VCBS middleware. With the eXtensible game description language (xgdl) we propose a standardized way to describe the virtual context of a gamer. VCBS is an interface between games and other internet applications, which involves game communities and supports gamers.

1. Introduction

Multiplayer Online Games are a thriving market. Millions of gamers from all over the world meet in the virtual worlds of Massively Multiplayer Online Games (MMOGs), competing against each other, forming teams or larger groups, and building social networks. MMOGs are designed for multiplayer game play, there thousands of players share a single game world and the virtual characters of the players can interact with each other at anytime. Especially in the persistent worlds of MMOGs people do not just play, they become participants in the games and a part of the game world and game community.

Multiplayer Online Games (MOG) live through participation and the communities gamers create. Exchanging information within communities about the game, its evolving state, certain events, and organized activities is extremely important. Therefore gamers build teams and social networks and thus exchange information

with state of the art (e.g. web 2.0) tools (e.g. [1], [2]). Examples for such tools are wikis with guidelines for a game, forums where tactics are discussed, websites and blogs where teams present themselves and their members. All these activities are initiated by the game community. Also, multiple tools are used by the community to enhance teamwork and collaboration (e.g. shared calendars or voice communication tools). All these tools and activities are separated from the game itself. There is no information exchange possible and the information which is provided by the community cannot be accessed in-game and vice versa. So the game communities shape another virtual space around the game including meeting points, discussion platforms, knowledge bases, this is called the environment of a game.

Making in-game information available allows the improvement of community activities and tools. MMOGs partially provide in-game support for organization of players in clans or guilds, but this only covers a small amount of the community's needs. To Maintain a group of players additional organisation is needed (e.g. to recruit new members, coordinate events or distribute duties). If it is possible for a gamer to present his gaming achievements on a website he can show his status to other gamers in the community. This for instance makes it much easier to find new members for a team.

In order to enhance MMOGs, support for gamers, and community participation we propose to use Virtual Context Based Services (VCBS). This concept enables information exchange between a game and its environment and the integration of services (e.g. communication services, cooperation services or information services). External services are integrated directly into a game based on the virtual situation of the gamer. With Virtual Context Based Services a guideline service, for example, can be used inside a game and there information according to current tasks can be retrieved directly. Another example is a virtual context based voice communication service. When a gamer joins an in-game group this services can automatically connect the gamer to the group's communication channel. Currently, a gamer needs to exchange the login data for a third party communication tool via in-game text chat and then leave the game, switch to the voice communication tool, connect to the server, join the right channel manually, and after the voice communication is established switch back to the game. Furthermore, services can utilize information a gamer provides during his

presence in the game world. With a virtual location based annotation service for example a gamer can annotate locations in the virtual environment and his annotation can then be shared with other gamers or may be provided in a wiki outside the game. We have developed a VCBS middleware which enables the integration of external services into a game. The external services can be developed and provided by the game community. Thus the creative participation is enhanced and influence opportunities for the game communities are provided.

Section 2 describes gaming communities and approaches currently used. In Section 3 introduces the concept of virtual context based services and it is described how an information exchange between a game and its environment can be accomplished. Section 4 describes the basics of virtual context and virtual parameters and introduces the extensible game description language. Section 5 describes our VCBS middleware and its architecture. Section 6 gives examples of Virtual Context Based Services and Section 7 summarizes the article.

2. Gaming Communities and Applications Used in Addition to Games

The development of communities has always been important in the area of computer gaming. The grouping of gamers in teams and the organisation of competitions developed over the years and today these game communities influence the whole online gaming market. The dedication of gamers to their games is generally very strong with the interaction and communication between gamers being essential elements, especially in MOGs. Gamers meet online for competitions, they support each other, organise themselves in teams, or build whole knowledge bases for games. Additionally, the possibility for gamers to have influence on design, appearance and control of a game gets more and more important.

All activities which do not take place inside the game, but are related to the game, can be found in the game environment (see Figure 1). The game community creates and maintains this environment using several tools and applications provided by the gaming industry, third parties, or the community itself, examples are knowledge bases (GuildWiki [1]), web pages of teams, clans, and guilds (MTW [2]), data archives (THOTTBOT [3]), or communication tools (TeamSpeak [4]).

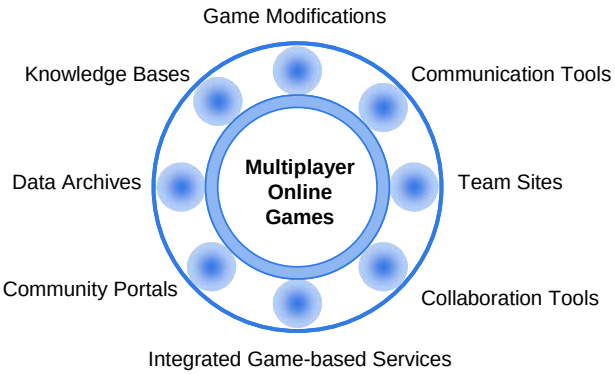

Fig. 1 MOG Environment

Such a game environment can be found for every MOG. Beyond the activities in the game environment, those games providing influence opportunities gain an extra benefit from the creative participation of the community. Some games include map editors where the community can design new levels and therewith enrich the variety of the game. The games industry also has begun to understand the creativity and developing power the community possesses. New titles such as Spore [5] or LittleBigPlanet [6] have been created with the intent of being developed by their communities. Thus, the content provision in these games shifts from the game developer to the game community.

Another approach are user created add-ons allowing gamers to customize parts of a game. For example, World of Warcraft (WoW) [7] offers the possibility to modify the User Interface (UI) through game add-ons. Although the changes that can be made to the UI of WoW have only a minimal influence, the community is very active creating various add-ons for the game (the internet portal curse.com [8] alone offers nearly 5000 WoW add-ons the community has created). The ideas of some of the more popular and helpful add-ons have been adopted by the developers and integrated into the official game through update patches.

However, the game worlds of MOGs are still isolated applications and no information exchange between a game and its game environment is possible (see Figure 2). In a current setting a gamer who seeks additional information or a guideline to a problem he is facing in a game, must leave the game, switch to his browser, find a website or wiki with the information, retrieve it, and then back switch back to the game in order to continue playing. This is cumbersome and a major interruption to gaming flow.

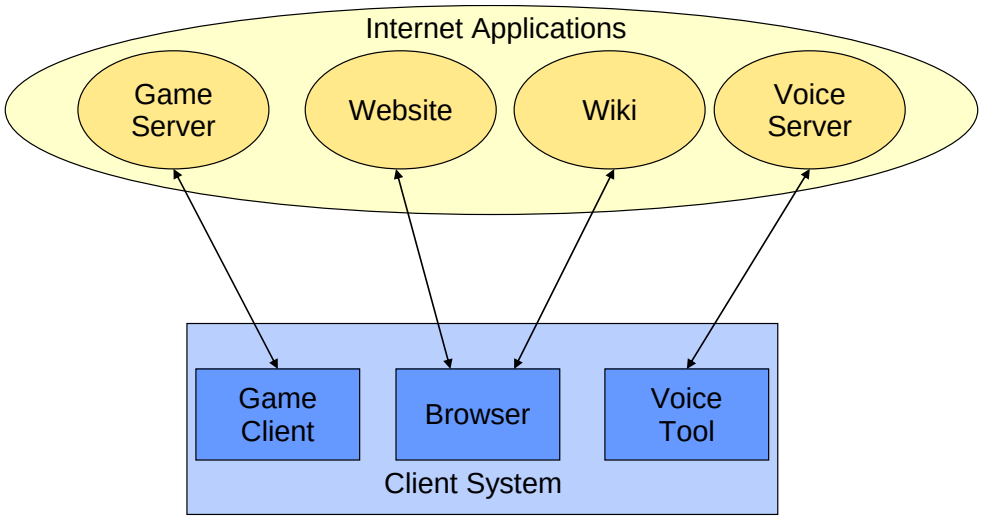

Fig. 2 Different Applications used in Addition to Gaming

First attempts are being made to overcome this shortcoming (e.g. developer driven game data presentation, launch platforms, or web portals (see [9])). Developer driven game data presentation is the presentation of in-game data on special websites provided by the game developers or publishers. On those websites certain information from a game and its players is presented. There the game developers utilize an unidirectional connection from the game servers to the presentation website. The gamers cannot influence which kind of information is provided and cannot add their own information. Two examples of developer driven game data presentation are the statistics websites of “Enemy Territory: Quake Wars” [10] and the “World of Warcraft Armory” [11]. The Enemy Territory: Quake Wars statistics website includes player statistics such as usage of classes or weapons, win/loss ratios, or game achievements. The World of Warcraft Armory provides information about characters (arena teams, guilds, or character equipment and skills) and the game world (items, dungeons, NPCs).

Launch platforms combine different applications. They can include server browsers (to find suited servers in FPS or RTS games), communication applications (chat clients), or buddy lists (to organize fellow gamers). They can monitor applications on the client system so they can realize when a game they support has been started by the gamer. Examples for launch platforms are ESL Wire [12], Steam [13] or Xfire [14].

In contrast to developer driven game data presentation are web portals hosted by third parties. Web portals don’t have direct access to information from the game servers, but use information presented on developer websites (e.g. from the World of Warcraft Amory). They mainly provide social networks for gamers (e.g. friend

lists or discussion groups) and give gamers a place to present themselves and customize their own profiles (e.g. by using blogs). Some web portals utilize additional information which can be obtained via game add-ons. Using such an add-on in-game information can be obtained from the game client and after the character has left the game, transmitted to the web portal. Examples for web portals utilizing this opportunity are buffed [15] and xchar [16].

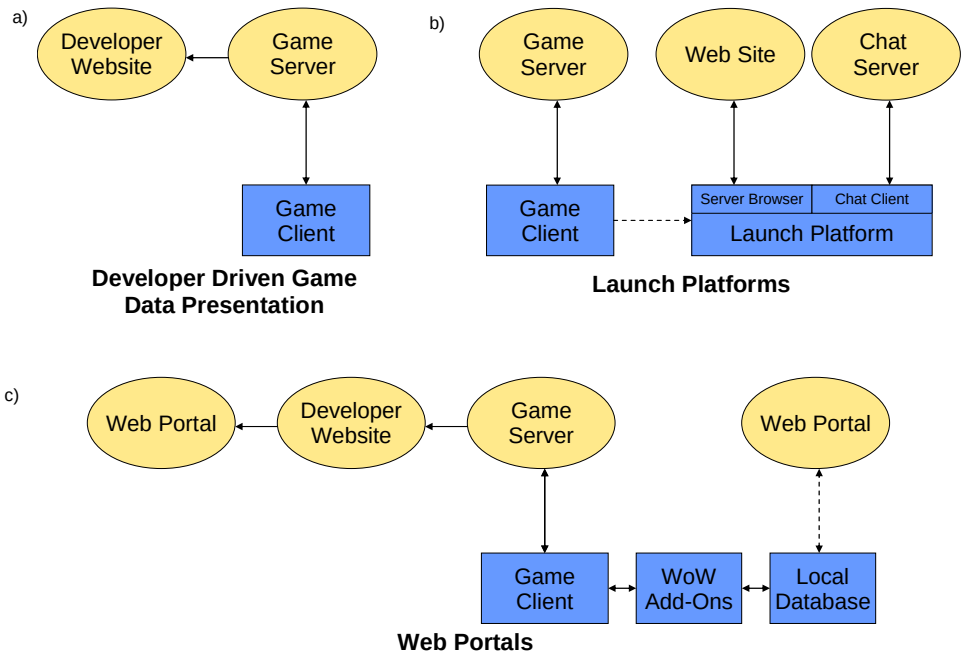

Fig. 3 Data Flow a) Developer Driven Game Data Presentation b) Launch Platforms c) Web Portals

Figure 3 shows the data flows of the previously introduced applications. The presented examples show first approaches which have been developed so far. No generic solution exists, suited for MOGs. Two important issues of future MOG development need to be solved: information exchange between games and their environment and inclusion of the participation of the community into the games.

Here arises an important opportunity, but also a significant challenge. The goal is to connect things that happen in-game with the game environment and vice versa and to include the accomplishments of the game communities into the game. To offer unlimited information exchange and community participation an unregulated access to MMOGs is needed, but then abuse of information would be possible. In order to prevent cheating, use of forbidden aids or unfair advantages, most of the games do not allow any connection to the game environment or the retrieval of in-

game information from the outside at all. Thus, new attempts are needed to realize a regulated opening of MMOGs.

3. Virtual Context Based Services for the connection of game and game environment

To enable a common information exchange between MOGs and their environment, there is need for a generalisation of interfaces and communication mechanisms. We have developed the Virtual Context Based Services middleware (VCBS middleware) providing an interface for exchanging virtual context information and external information. With the VCBS middleware different applications can provide and utilize information through services (see Figure 4). The external services can be created by game developers or the game community.

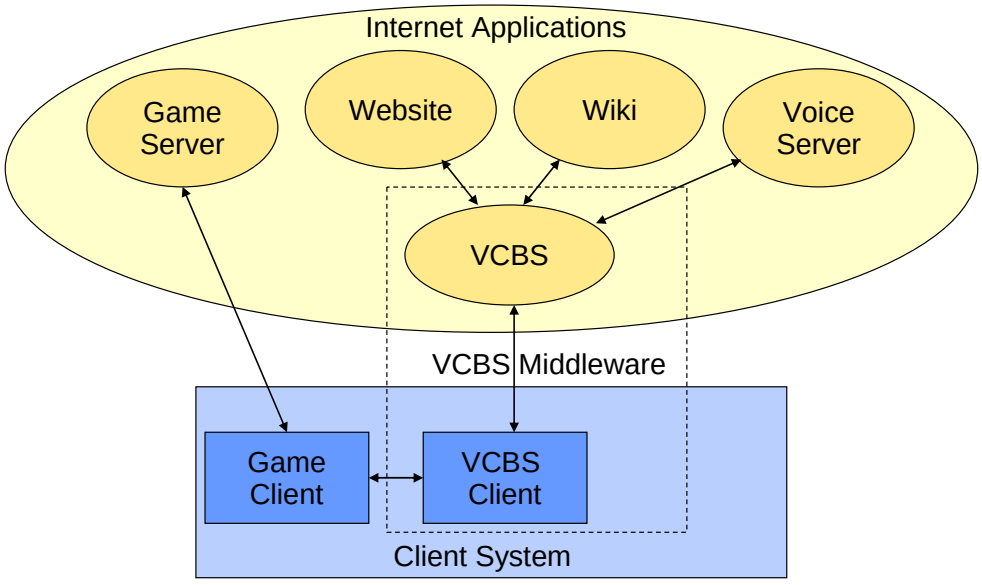

Fig. 4 VCBS Middleware - External View and Abstract Information Flow

To connect a game with its environment two directions of the information flow have to be considered. First the use of in-game information in the game environment (game to environment) and second provision of out-game information within the game (environment to game).

In order to provide in-game information about a gamer his *virtual context* has to be taken into account. “A context is an abstract and meaningful description of the relationship between objects and their environment” [17]. Applied to virtual environments the virtual context describes the relationship between virtual objects (including the virtual characters of the users) and the virtual environment (virtual

world). This *in-game context information* can be collected and then made available to the game environment where it can be accessed by various applications (e.g. for online platforms to display actual in-game information, like the current position of a user in the game world). With VCBS also additional information provided by the gamer in the virtual environment can be provided associated with his virtual context (e.g. to create virtual location based annotations). The information flow from the environment to the game can be realized by providing access to *external information* in-game (e.g. guidelines the community has collected in a wiki).

In connecting both directions of the information flow the external information, which is provided in-game, can be adapted to the virtual context of the gamer (see dotted arrow in Figure 5). Certain information, the gamer might need in his current situation in the game, is provided in this manner (e.g. a guideline for the current task of the gamer or additional information about the area the gamer's character is currently located in). Figure 5 shows an abstract visualisation of the information flow between a game and its environment.

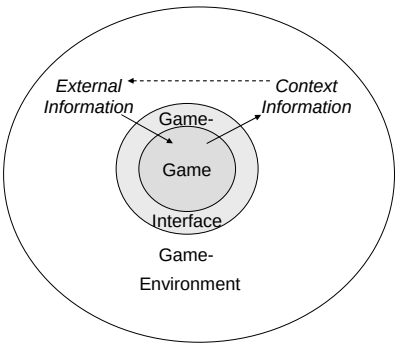

Fig. 5 Information Flow between Game and Game Environment

Thus, both information flows are considered, the in-game context information which is provided in the game environment can be utilized by the game community and external information the community provides in the game environment is made available in the game.

In order to achieve a generic solution certain challenges need to be addressed. Those challenges are the definition of virtual context and how virtual context can be described formally, the definition of interfaces for MOGs, capturing and transmission of information and of the descriptions of the virtual context as well as the integration of external information into a game. Additionally, a filtering mechanism to regulate the information flow has to be integrated.

Defining virtual context and a formal description of virtual context which are the foundations for the interconnection between game and game environment and constitute the base for generic game interfaces are presented in the following section. Subsequently information transmission and integration are introduced.

4. Virtual Context Description

To describe the virtual context of a gamer his virtual alter ego (avatar or virtual character), which is the gamers representation in the virtual world, has to be analyzed. This virtual character and the interaction of the character with the virtual world determine the virtual situation of a gamer. This virtual situation can be described through virtual context likewise context in the real world can be used to describe the situation of a person.

4.1 Context Based Service Provision

A virtual character has a certain set of interactions he can perform [18]. This depends on the class of the character and how he is skilled (during the development of a virtual character it is possible for a gamer to specialize his character and to acquire new skills). The set union of all interactions a virtual character can perform with the virtual world defines his *principle interaction opportunities*. But the character can not perform every interaction at any time (e.g. to use a swimming skill there has to be deep enough water).

The interaction a virtual character performs in a certain situation depends on influence factors. Those influence factors are the *virtual* and *real parameters*. Virtual parameters (such as equipment, location, or status of the character) describe the virtual context a virtual character has in a current virtual situation. Real parameters (such as goals, availability, or experiences of the gamer) are user dependent and describe the gamer's real context. The virtual situation of a character reduces the set of interactions he can perform in his current situation to the *actual interaction opportunities*. Finally, the gamer decides based on his real context which interaction is appropriate for him and chooses the *actual performed interaction*.

Likewise the principle interaction opportunities all possible services could be offered to a user at any time. The idea of Virtual Context Based Services is the provision of services regarding the virtual situation of a gamer's character. Thus,

only those services suitable to his current situation are offered to him (see Figure 6).

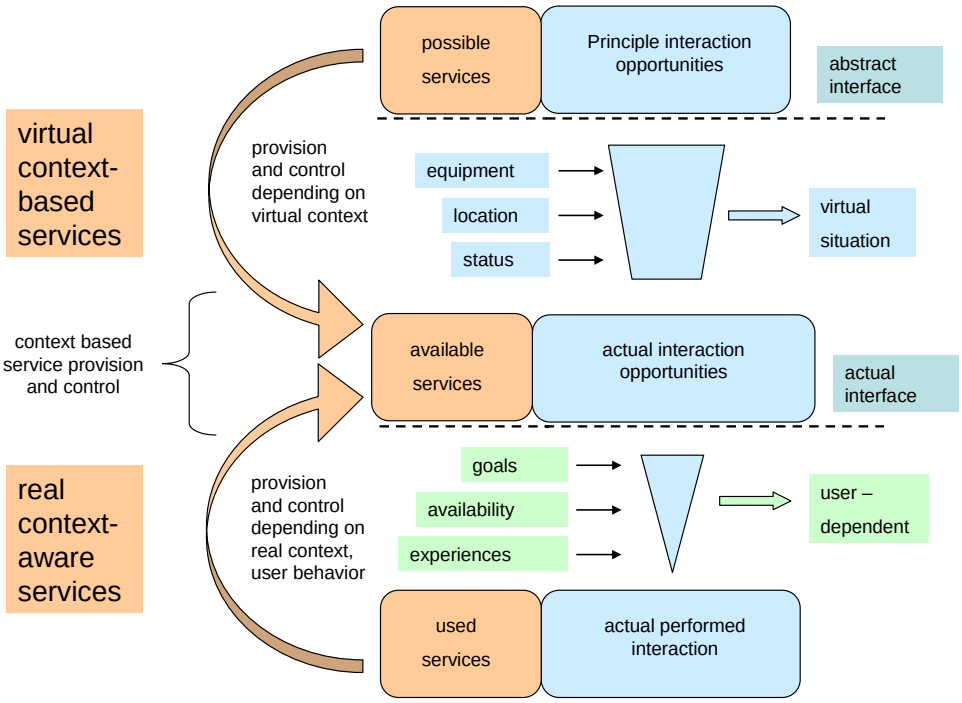

Fig. 6 Abstract VCBS model

Figure 6 visualizes the connection of services and interactions. The *principle interaction opportunities* of a gamer are reduced by his current *virtual situation* to a set of *actual interaction opportunities*. The *virtual situation* incorporates virtual parameters such as, *equipment*, *location* and *status*. The *principle interaction opportunities* correspond to all *possible services*. Regarding the *actual interaction opportunities* only a subset of all services is suited for a given virtual situation. For virtual context based services the *available services* correspond to the *actual interaction opportunities*. Likewise the *actual interaction opportunities* are reduced by *user-dependent* parameters of the gamer (such as, *goals*, *availability*, and *experiences*) to the *actual performed interaction*. Accordingly it depends on the user which of the available services he will use. So for context-aware services *used services* correspond to the *actual performed interaction*.

In addition to the virtual context based services real context-aware services can be used to tailor services to the user behavior. Both ways of context depended service provision can be used to provide services tailored to a user's behavior, preferences and current situation. This work focuses on virtual context based services. There are still many open research issues in this area. In future work we want to enhance

virtual context based service provision by means of context-aware service provision to improve the set of available services for the gamer.

Using context information to provide services is an issue in different application domains. Commonly real context parameters collected by sensors are used to provide context aware services (e.g. [19]). Context-aware services are already used in different scenarios (e.g. communication scenarios [20], [21] or computing applications [22], [23]). The usage of virtual context based services is still actual research and the VCBS middleware is the first implementation in this field.

4.2 A Description Language for Virtual Context based on Virtual Parameters

Based on the virtual context and its description data model, information exchange between a game and other internet application can take place. The game description can be used to define abstract interfaces for information exchange. This enables provision of in-game information to other applications or inclusion of external information into a game. Virtual Context Based Services utilize this data flow.

To use virtual context for the provision of services the virtual parameters needs to be standardized and formalized. However every MOG may have its own specific virtual parameters thus the main challenge in creating generic VCBS is to have a generic parameter set. We have developed a generic base set of virtual parameters valid to describe virtual context. This base parameter set is the foundation for the description of virtual context in MOGs and the information exchange between a MOG and its game environment. The base parameter set is a subset of virtual parameters, which can be generalized for MOGs. Using the base parameter set (MOG base parameter set) we have provided a generic data model, but it is also important to include genre or game specific characteristics in the description of virtual context. Thus, we have designed the description language to be extendable for different MOG genres and further for single games. The extended parameter sets always include the more generic ones to allow interoperability and a shared common base.

We have divided the parameters into four parameter categories: general, location, character, and status. The general parameter set includes general information about the player (e.g. player name), the game time (e.g. overall time the player has

played the game) and server information (e.g. server name). The location parameter set includes position information (e.g. coordinates of the character). The character set is about the virtual character of the player including specifics of the character (e.g. character name) and his equipment. The status set contains information about the current state (e.g. current action) and about grouping (e.g. if the character belongs to a group). Figure 7 shows the virtual parameters that can be applied for MOGs.

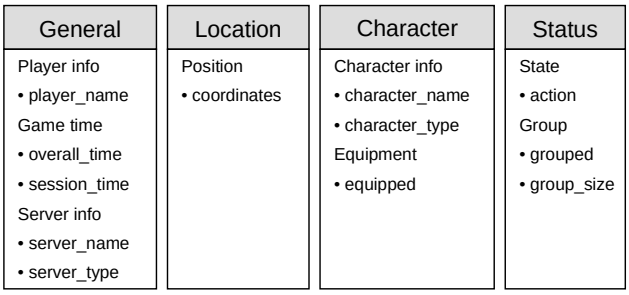

Fig. 7 MOG Base Parameter Set

The base parameter set can be extended to cover the specifics of different MOG genres (e.g. for MMORPGs or FPS see Figure 8) and even further to describe specific games in detail. The resulting parameter sets always include the base parameter set to ensure interoperability. The different parameter sets are needed to cope with the specialized needs of the different genres and games. For example, the location information in the FPS parameters set includes additional information about the map a character is currently located on and the MMORPG parameter set includes additional information about the area a character is currently located in.

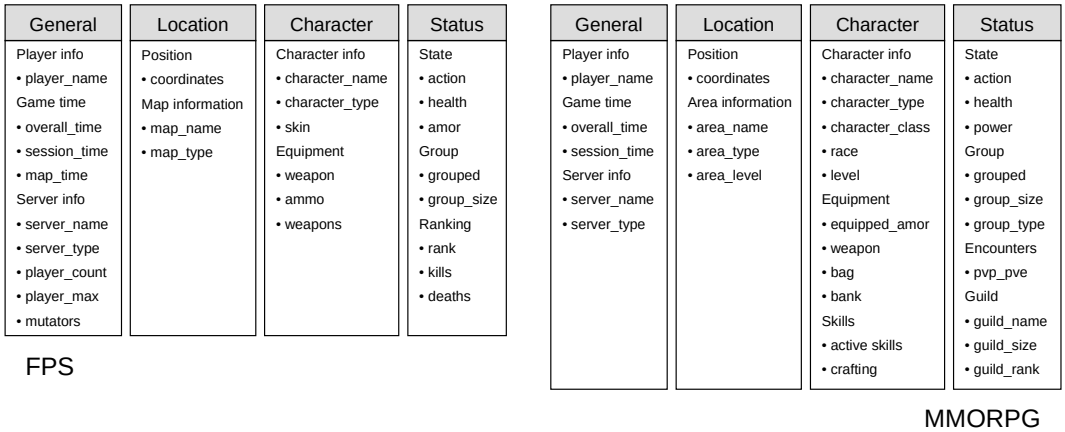

Fig. 8 FPS and MMORPG Base Parameter Sets

Examples for virtual parameters are the position coordinates of the character in the virtual environment, his equipment (e.g. weapons, armor and ammunition), or his grouping status (if he is a member of a team, group, or another in-game institution). The parameter set is instantiated by a game and interpreted accordingly. The game specific interpretation of a parameter can be quite similar for divers MOGs (e.g. the name of the player) or it can differ (e.g. the equipment of the character or his state). A good example for a differing interpretation is the equipment in Role-Playing Games (RPGs) in contrast to the equipment in a Real-Time Strategy (RTS) game. In a RPG a virtual character has certain clothing and accessories with which he is equipped, the equipment in a RTS game may include the buildings that are under the character's control or with which he may be associated.

However by finding a generalization, as we have done using the base parameter set, which is the common basis for all MOGs, an abstract data model can be defined. Based on the parameter sets we have developed a virtual context description language, the eXtensible Game Description Language (xgdl). The xgdl is based on XML and defines virtual parameters, which can be generalized for different MOGs and describe the virtual context. The xgdl can be extended to include additional virtual parameters and in addition due to the modular concept it is also possible to include further parameter sets. The XML description offers a well defined way to exchange in-game context information. Thus the virtual parameters and the parameter sets are the base for interface definition between games and other internet applications.

5. A Middleware for Virtual Context Based Services

In order to provide a generic solution for the information flow between games and their environment and to overcome the shortcomings discussed in Section 2 we have developed the VCBS middleware. The VCBS middleware is a novel approach to define a standardized connection between a game and its environment.

The VCBS middleware is based on Virtual Context Based Services. The goal of context based services is to provide services to a user regarding his current situation. Virtual context based services correspond to the virtual context of a user and thus collect information about the virtual context of a user's character. Virtual

parameters like location, experience, or equipment describing the virtual context can be used to select services which are suited to the user's current virtual situation. Those Services, which are appropriate for the virtual situation of a virtual character, are offered to the user. For example a service supporting group communication is offered to the user when he joins a group. The user can then decide whether to use a service or not. This is an important aspect of the VCBS. Gamers using VCBS services decide about service usage including what information about their virtual character is made available. For example if a gamer does not want to show his current in-game location to other users he will not use that kind of service. In-game information forwarded to the external services is constricted. Prior to service usage the gamer should be able to check which in-game information will be utilized by the service. Additionally information about service should be also available during service usage und some kind of service control mechanisms is needed. We have developed the Service User Interface (SUI) which is an extension to the user interface of the game. It includes different functions and is composed of different drop-down menus (see Figure 9).

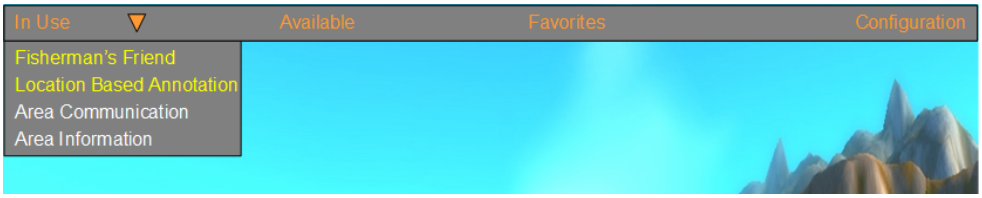

Fig. 9 Service User Interface (SUI)

The category *In Use* includes all services a gamer is currently using and provides additional information about them. *Available* includes all currently available services. Services a gamer wants to use any time they are available can be added to his *Favorites* so they will be automatically confirmed for him. The *Configuration* category includes options configuration of the service user interface such as appearance configuration and an ignore list including services the user never wants to use.

Services are offered to the gamer when they are available, except when they are on his ignore list. In most cases a service, which a user wants to use, must be first searched for and thus services that the user does not know about will only be found by chance. If a service is offered to the user, he knows that the service exists and then he is free to decide whether or not to use this service.

The main goals of the VCBS middleware are to connect games with their environment and to enable community participation. The services are free to be developed by the community. The community can develop services which are then connected and integrated to the game with the VCBS middleware. This way new participation opportunities for the community are created. Additionally the VCBS middleware provides the same service interface for all MOGs so a gamer does not have to learn new tools, APIs, or programming languages every time there is an interest to create a service for a new game. In providing a service that is free of fees to the game community solutions can be found that are commonly used in other application areas (e.g. portals to sell and buy cars). The free service will then be made possible by covering the expenses through the collection of fees from professional service providers that use the VCBS middleware.

An overview of the information flow between a game and its environment through the VCBS middleware is given in Figure 10. The in-game context information is forwarded by the VCBS middleware and provided to the applications in the game environment. Information provided by external services in the game environment is transferred back and integrated into the game.

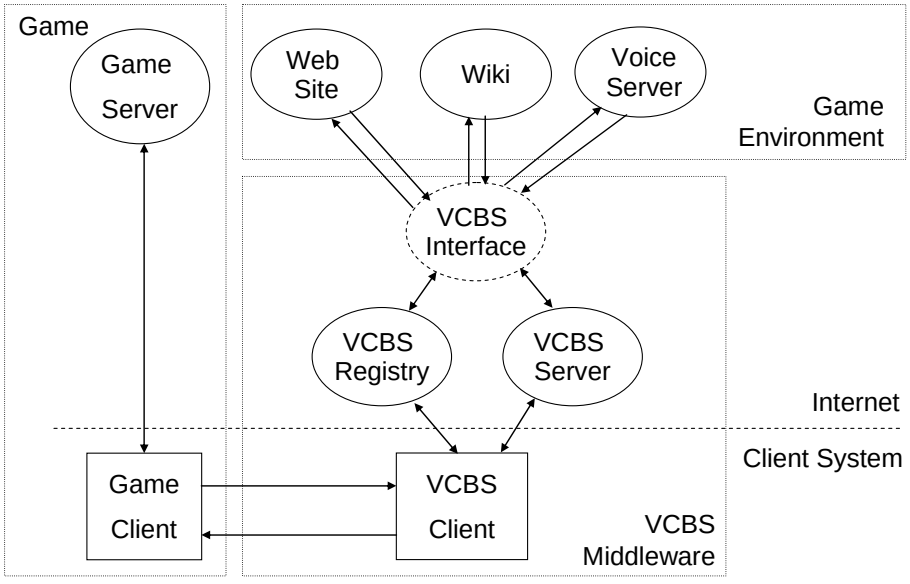

Fig. 10 VCBS Middleware – Internal View and Information Flow between Game and Game Environment

The VCBS middleware includes capturing and processing of virtual parameters, virtual situation based filtering, forwarding of in-game context information, and integration of external information into the game. The middleware provides an

interface for external services, which is based on the xgdl game description language.

5.1 Virtual Situation Based Filtering

One big issue in gaming research is the prevention of cheating ([24], [25]). To provide all context information anytime independently of the virtual situation would enable abuse of information. For example, in a competition between gamers it would be an unfair advantage if the position information of the gamer's character was available to the other gamer(s) to see. It would be an unfair advantage to the gamer's opponents as he will thus always be visible to them. To prevent the abuse of in-game information and to be able to provide information which is harmless, the forwarding of in-game information is regulated by the VCBS middleware. Instead of allowing his position to be visible anytime, this information is provided only after any virtual combat, when it is harmless and helpful, maybe in the pursuit of friends,

A regulation of the information flow can be realized by including a filtering mechanism for in-game information based on virtual context. Virtual context defines the virtual situation therefore regarding the virtual context of a gamer's character it can be decided which information about his current situation should be forwarded to the game environment.

This situation dependent information filtering is based on rule sets. The rule set can be defined by the game developers which then can be integrated into the VCBS middleware. It is nessesary to provide such an influence opportunity for the game developers because they are responsible to keep their game fair and balanced. If any in-game information is accessible at any time fairness can no longer be guarantied causing harm or even ruin of game and game play. The rule sets are also based on virtual parameters and can be defined in the same granularity as the game descriptions. So on one hand for any game it is possible to define rules that ensure that information which could be used to the disadvantage of the gamer is not accessible. On the other hand in-game context information can be made accessible any time else. An example rule allows that the virtual position of a gamer's character is only accessible if the character is not in combat with other gamers. Therefore, depending on the situation, whether or not the character is not in combat (or is fighting against the virtual environment, pve - player versus

environment) or is fighting against another character (pvp - player versus player) will determine whether or not his in-game position will be forwarded to the game environment.

5.2 VCBS Middleware Architecture

The VCBS middleware supports both directions of the information flow. It defines an interface to the game for the capturing of in-game parameters (virtual context information). The captured in-game information is provided to the game environment where it can then be accessed by various applications. And it provides an interface for the in-game provision of data from external services. Additionally, external services processing the virtual context information can provide data the gamer needs in his current situation inside the game through VCBS middleware. An example service which is using both directions of the information flow is a service supporting gamers fishing in the virtual world ("Fisherman's Friend"). The "Fisherman's Friend" service is offered to a gamer when he is standing near a virtual lake or river and equips a fishing rod. When the gamer is using the service, information about his fishing trip, such as the fish he catches, is transmitted to the external service database. On the other side the service derives statistical information from the captured values and transmits this additional information, which is then offered to the gamer. Now the gamer can check what species can be caught and with which probability, if the waters is suited for his fishing skill, and what other fishing sites might appeal to him.

The architecture of the VCBS middleware consists of three base components (see Figure 11) [9]: an interface to the game client and to the user (**VCBS Client**), servers for information processing, regulation of the information flow and forwarding of in-game information (**VCBS Server**) and a management component (**VCBS Registry**). Also the VCBS middleware defines an interface (**Service Interface**) for external services.

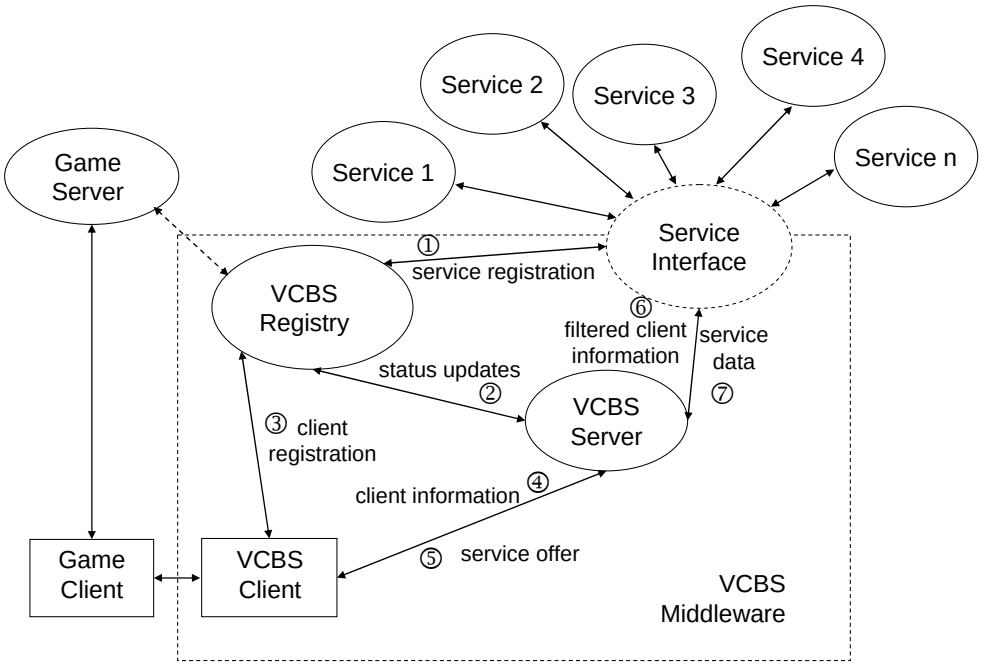

Fig. 11 VCBS Middleware Architecture Overview

The **VCBS Registry** performs the administration tasks of the middleware. It manages the users of the VCBS middleware, the different VCBS Servers and all registered external services. To use a service as "Fisherman's Friend" with the middleware it has to perform a *service registration* (1). Once registered, a service is distributed to the VCBS servers and thereafter can be offered to the VCBS users. The VCBS Registry distributes the active users to the different VCBS Servers to do load balancing and ensure the scalability of the middleware. The Servers are updated by the Registry using *status updates* (2).

The **VCBS Client** utilizes a connection to the game and provides an interface for the user. It is located on the user system. The VCBS Client performs the registration and authentication of the user with the VCBS Registry (*client registration* (3)). Our vision is that game developers use a common interface for exchange of virtual parameters and external information, but unfortunately such common solutions are hard to accomplish. Thus, to connect the VCBS Client to different games we are currently using game plug-ins. These plug-ins facilitate connections with different APIs used by game developers and provide exchange of virtual parameters and external information. The VCBS Client is implemented in a modular way and easily extendable. The VCBS Client captures and transmits the virtual parameters to a VCBS Server (*client information* (4)). The VCBS Server checks whether the service the users currently uses are still valid and if new services have become valid based on the client information. If a new service

is available a *service offers* (5) is send to his VCBS Client. For example a user has changed his location and is now standing beside a virtual lake and his equipment contains a fishing rod that means he now meets the requirements for the "Fishermen's Friend" service. Some time later in a client information update the user is still standing at the same location but exchanged his fishing rod with a sword, then the VCBS Server informs the VCBS Client and negates the respective service offer.

A **VCBS Server** uses in-game information to determine services suited to the users virtual context. When the user approves a proposed service the VCBS Server forwards the *filtered client information* (6) to the external service if the required parameters are valid regarding the rule sets for situation based filtering. The forwarded information depends on the service, i.e. for the "Fishermen's Friend" service virtual parameters including location, equipment and loot information are required. Then the external services can utilize the in-game information (e.g. calculate fishing statistics out of the virtual information), can present this information (e.g. on a website) and can send *service data* (7) to the VCBS Client. The VCBS Client then integrates the external information into the game. For "Fisherman's Friend" the external information which is integrated into the game by the VCBS Client contains the name of the current fishing site, information about fish which can be caught there, and the difficulty of the waters (see Figure 12). Expanding the given information to provide more or more detailed information is easily possible.

Fig. 12 Information Provision of the "Fisherman's Friend" Service

When the virtual situation of the user has changed and a service is not appropriate anymore, the VCBS Server stops forwarding the in-game information to the service.

6. Examples for Virtual Context Based Services

In the following section we show two examples of Virtual Context Based Services. We have implemented those examples and tested regarding functionality and usability. As a proof of concept for our generic approach we choose two utilization examples that are game independent. The virtual location based annotation service is appropriate for MMOGs with huge worlds. Our second example the virtual context aware dynamic voice communication service is in addition genre independent.

6.1 Virtual Location Based Annotation

MOG worlds are huge. There is plenty of virtual space to be explored by gamers and even teams of gamers spend much time to completely discover certain areas of continents that are part of the virtual world. To effectively discover the world teams use wikis and other documentation tools to share the advances of the team members. With Virtual Context Based Services it is straight forward to collect user annotations within a game, merge this information with the user's location context and provide it to an external service like a wiki. The service application stores the annotations of all users using this service functionality in the game. We have developed a location based annotation service on top of the VCBS concept using the VCBS middleware, a World of Warcraft add-on and a simple web service (the location based annotation service) to store and share annotations.

In World of Warcraft the gamer can describe a situation or object by annotating his current location. A small window pops up where he can write down a text annotation. This window also shows his current location. The VCBS client captures this information and converts it into a valid xgdl context description. The xgdl description is processed by the VCBS Server, which sends it to the location based annotation service together with the gamer's annotation. The service stores the annotation and links it to the user and his friends. The next time he or his friends approaches the virtual location of an annotation a window pops up showing the description of the situation or object which was annotated at this location (see Figure 13).

Fig. 13 Virtual Location Based Annotation Service

The location based annotation service is a simple asynchronous service provided for gamers. Beside using it for a “documentation of the virtual world” it can be applied for other use cases for example in paper chase events within the virtual world.

6.2 Virtual Context Aware Dynamic Voice Communication

For team communication in real-time voice communication is used because chat massages are not feasible for time critical game situations. Normally external tools are used for voice communication which support static chat channels for more than 40 players. Static chat channels and their dedicated participants can not be reorganized while playing a game without switching to the voice client.

In [26] we introduced a concept for dynamic voice communication for competitive and tactical team-play in Multiplayer Online Games based on templates. With a template a team leader can pre-plan common situations for upcoming tasks (see Figure 14). The game dependent communication hierarchy is shown at top of Figure 14. In the planning phase the team leader decides to focus on two possible template configurations (see b) and c)). The fallback configuration a) matches the static channel configuration, providing a single chat room including all team members. The planned tactical variants b) and c) show the benefit of the dynamic voice communication concept. Both configurations introduce sub-team leaders and the creation of dynamic channels. In the game the team leader can switch through the templates and thus distribute the team members to different chat rooms according to the current game situation.

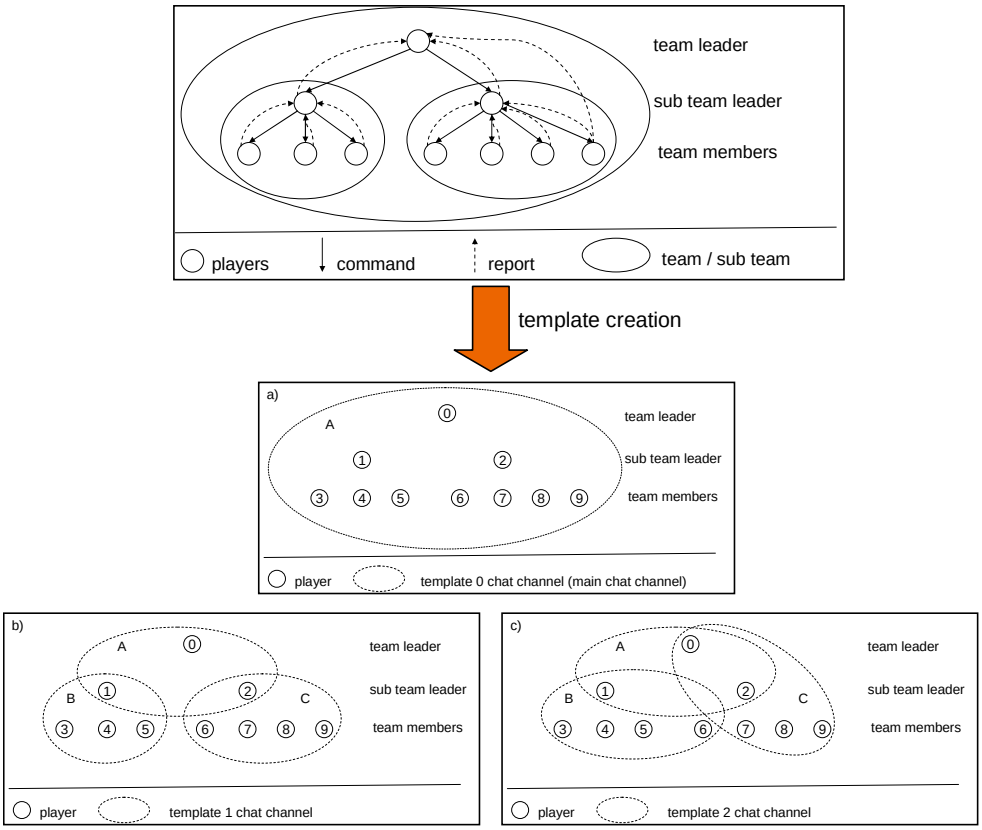

Fig. 14 Situation Dependent Player Distributions with Dynamic Voice Chat Channels

This allows more parallel communication than in single static chat channels. With Virtual Context Based Services a service for dynamically managing of chat channels based on the virtual context of the gamer is possible. The virtual context aware dynamic voice communication service can automatically add friends seen in the virtual world to the own chat channel, distribute team members automatically into different chat rooms or realize a location based sound and volume control in the virtual world.

7. Summary

In the future development of Massively Multiuser Online Games gaming communities and the enhancement of gaming experience are important issues. MOGs live through gamer participation and growing communities. Especially in the MMOG market which is highly competitive, the binding of a critical mass of users to a game is not only important for its growth, but determines its success or failure [27]. The support of community development in the game [28] and in the game environment improves the binding of the user to the game and thus by itself one success factor. Only MMOGs with strong communities can be successful in the end.

Game developers and publishers begin to understand that game communities mean formidable additional power to the development of the virtual world they provide. To utilize a creative participation of the community and to establish an information exchange between games and their environment a generic game and genre independent solution is required. With our concept of Virtual Context Based Services and the implementation of the VCBS middleware we provide a solution for both of these issues.

We have analyzed Multiplayer Online Games and game environments to develop strategies how gamers in virtual worlds can be supported and an information flow for supporting community participation can be realized. To support a gamer in a virtual environment his virtual situation has to be taken into account. This virtual situation is defined by the virtual context of the gamer and can be described using virtual parameters. We have developed a description language for virtual context based on virtual parameters, the extensible game description language (xgdl). With xgdl we can completely describe any context of a gamer in a MOG from general information up to game specific complex information. Based on the description of virtual context Virtual Context Based Services can be realized and services supporting users in virtual worlds can be applied to MOGs.

To support the information flow between a game and applications in the game environment we have developed the VCBS middleware. The VCBS middleware provides consistent interfaces to connect games and external services and situation dependent user support. The middleware facilitates the transparent transport of information in both directions. Multi-purpose services can be created to be used with the VCBS middleware as the examples in section 6 show. Using a middleware enables the integration of community created services into a game and provides means for creative community participation. Regarding the virtual situation of a gamer the service can be customized and information provided while a gamer is playing in the virtual world of a game can be utilized in the game environment. Additionally current services like third party voice communication systems can be improved.

Using the VCBS middleware improves support for gamers, minimizes interruptions in the gaming flow, and additionally increases the participation of the gaming community in the adding to the collective gaming experience. Thus,

the VCBS middleware adds a high value to the further development and improvement of Massively Multiuser Online Games in the future.

8. References

1. Guildwiki: a wiki guide for ArenaNet's Competitive Online RPG, Guild Wars, http://guildwars.wikia.com/wiki/Main_Page, last accessed 17.12.08
2. MTW, http://www.mymtw.com/cms/front_content.php?idcat=12, last accessed 17.12.08
3. THOTTBOT: World of Warcraft Database, http://thottbot.com/, last accessed 17.12.08
4. TeamSpeak Systems: TeamSpeak, http://www.teamspeak.com/, last accessed 17.12.08
5. Maxis, Electronic Arts: Spore, www.spore.com, last accessed 17.12.08
6. Media Molecule, Sony Computer Entertainment: LittleBigPlanet, www.littlebigplanet.com, last accessed 17.12.08
7. Blizzard Entertainment: World of Warcraft, www.worldofwarcraft.com, last accessed 17.12.08
8. Curse.com, http://www.curse.com/, last accessed 17.12.08
9. Bergsträßer S., Hildebrandt T., Rensing C., Steinmetz R. A Middleware for the Controlled Information Exchange between Online Games and Internet Applications. In: 16. ITG/GI - Fachtagung Kommunikation in Verteilten Systemen 2009 - KiVS 2009, Kassel, Germany, March 2009
10. Activision: Enemy Territory: Quake Wars, Official PC Stats Site, http://stats.enemyterritory.com/, last accessed 12.01.09
11. Blizzard Entertainment: The World of Warcraft Armory, http://eu.wowarmory.com/, last accessed 12.09.08
12. ESL: ESL Wire, http://www.esl.eu/de/wire, last accessed 10.09.08
13. Valve: Steam, http://store.steampowered.com/, last accessed 12.01.09
14. Viacom: Xfire – Gaming Simplified, http://de.xfire.com/, last accessed 12.01.09
15. buffed.de: Das Portal für Onlinespiele, http://www.buffed.de/, last accessed 12.09.08
16. xchar: WoW meets real life, http://www.xchar.de/, last accessed 10.09.08
17. Görtz M.: Effiziente Echtzeit-Kommunikationsdienste durch Einbeziehung von Kontexten. PhD thesis, Technische Universität Darmstadt, 2005. http://elib.tu-darmstadt.de/diss/000592/.
18. 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
19. Schmitt J., Kropff M., Reinhardt A., Hollick M., Schäfer C., Remetter F., Steinmetz R.: An Extensible Framework for Context-aware Communication Management Using Heterogeneous Sensor Networks. no. TR-KOM-2008-08, November 2008.
20. Goertz M., Ackermann R., Schmitt J., Steinmetz R.: Context-aware communication services: A framework for building enhanced ip telephony services. In International Conference on Computer Communications and Networks (ICCCN), pages 272-279, October 2004.

21. Moschgath M.-L.: Kontextabhaengige Zugriffskontrolle fuer Anwendungen im Ubiquitous Computing. PhD thesis, Technische Universitaet Darmstadt, 2005.
22. Schilit B., Adams N., Want R.: Context-aware computing applications. In IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, US, 1994.
23. Singh A., Acharya A.: Using session initiation protocol to build context-aware voip support for multiplayer networked games. In Proceedings of 3rd ACM SIGCOMM workshop on Network and system support for games, 2004.
24. Chambers C., Feng W., Feng W., Saha D.: 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
25. Li K., Ding S., McCreary D., Webb S.: 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
26. Hildebrandt T., Bergsträßer S., Rensing C., Steinmetz R. Dynamic Voice Communication Support for Multiplayer Online Games. In: Netgames 2008, 7th Annual Workshop on Network and Systems Support for Games, p. 111--116, Worcester, USA, 2008
27. Strain, J.: How to create a successful MMO, Games Convention Developers Conference (GCDC), Leipzig, 2007, http://eu.guildwars.com/press/article/jeffgc2007/
28. Ducheneaut, N., Yee, N., Nickell, E., Moore, R. J: 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

Sonja Bergsträßer is working at the Multimedia Communication Lab at Technische Universität Darmstadt. She graduated in 2005 with a diploma in computer science. She is PhD student at Technische Universität Darmstadt and research training group fellow. Her research interests cover the areas of networked gaming and context aware communications with a focus on utilization of virtual context information and service provision in virtual environments.

Tomas Hildebrandt is a PhD student at the Multimedia Communications Lab of Technische Universität Darmstadt. In 2001 he graduated as Dipl.-Inform. from the Technische Universität Darmstadt. His research interests lie in the area of networked gaming, especially the technical challenges in the community driven environment of multiplayer online games lie in his focus.

Dr.-Ing. Christoph Rensing completed his PhD thesis in 2003 on "A policy-based access control architecture for the Multi-Service Internet". He works as research group head at the Multimedia Communications Lab at Technische Universität Darmstadt leading a team of researchers in the area of Knowledge Media. In addition he is research consultant at the Hessian Telemedia Technology Competence Center. He has published more than 60 papers in national and international conferences and journals. Dr. Rensing is a member of the ACM, and GI.

Prof. Ralf Steinmetz worked for over nine years in industrial research and development of distributed multimedia systems and applications. He has been head, since 1996, of the Multimedia Communications lab at Technische Universität Darmstadt, Germany. From 1997 to 2001 he directed the Fraunhofer (former GMD) Integrated Publishing Systems Institute IPSI in Darmstadt. In 1999 he founded the Hessian Telemedia Technology Competence Center (httc e.V.). His thematic focus in research and teaching is on multimedia communications with his vision of real ”seamless multimedia communications”. With over 200 refereed publications he has become ICCC Governor in 1999; was awarded the ranking of Fellow of both, the IEEE in 1999 and the ACM in 2002.