

GASP: an open source gaming service middleware dedicated to multiplayer games for J2ME based mobile phones

R. Pellerin+*, F. Delpiano^, F. Duclos§, E. Gressier-Soudan+, M. Simatic*
pellerin@cnam.fr, fabien@infraworlds.com, fduclos@bouyguetelecom.fr,
gressier@cnam.fr, michel.simatic@int-evry.fr

+ CNAM-CEDRIC, 292, rue St Martin, 75141 Paris Cedex 03, Tel: +33 1 58 80 85 13 France

^ InfraWorlds, 33 avenue de la République, 75011 Paris, Tel: +33 1 58 62 53 00 France

§ Bouygues Telecom, Centre d'affaires la Boursidière, 92355 Le plessis Robinson, Tel: +33 141 09 27 54 France

* GET-INT, 9 rue Charles Fourier, 91011 Evry Cédex, Tel: +33 1 60 76 45 61. France

ABSTRACT

GASP is an open source platform that conforms the Open Mobile Alliance (OMA) v 1.0 specifications. It is able to handle basic services for building and handling interactions between players such as session management, communications... GASP is implemented in Java. It offers a server and a client part. It is defined for MIDP and DOJA J2ME profiles. GASP is demonstrated through Kouiz, a multiplayer game for mobiles presented in the second part of this paper. During solo phases of the game, the player tries to breed animals called kouizs. The aim is to improve their skills. With improved skills, kouizs have better chances to win against challengers in the multiplayer arena. In this mode kouizs can fight. This paper ends by discussing our current results and what we plan to do next with the gaming service platform GASP.

Key Words

Multiplayer games, mobile phone, GASP, OMA

1.INTRODUCTION

The feasibility of multiplayer games over mobile networks is studied and tested since June 2003. First, in the context of MEGA [SIM04a][SIM04b], a project promoted and led by GET-INT, two aspects have been carried out. The first one was dedicated to game service middleware in a broad sense. It led to the first version of GASP, Gaming Service Platform, the library described in this paper. The second aspect was related to user services acceptance and human factors.

The JIMM project succeeded to MEGA. The results obtained in MEGA have been reused and extended in

JIMM. JIMM offers an implementation of a multiplayer game for mobile phones and an open source middleware for gaming.

JIMM is a French consortium composed of three partners. Bouygues Telecom is the leader of the project. As a mobile network provider it promotes i-mode cellular phones in the context of JIMM. It brings expertise on mobile network communications. CNAM-CEDRIC, as a research Lab on multiplayer games, is in charge of enhancing GASP. GASP is also a common research project with GET-INT. Infraworlds, as a game programming and editing company, develops a multiplayer game that benefits from GASP functionalities to handle multiplayer interactions. Kouiz is a demonstrator for GASP

This paper presents our work. Section 2 is devoted to Gaming Services: the OMA proposal, a state of the art, the services implemented by GASP. Section 3 presents the game Kouiz, its goal is to illustrate the use of GASP. Section 4 states our current results, the advantages of an open source middleware, and concludes.

1.GASP, A GAMING SERVICE

MIDDLEWARE

Principles for networking in multiplayer games are nowadays well known [SME01]. But when trying to implement these principles in the context of mobile, game developers meet new challenges: 1) Terminals have limited capabilities (for instance, the executables cannot be more than 100-200 Kb), 2) The sending of a message has a non-negligible cost for the player, 3) http is usually the only available protocol to

communicate with a remote server (either because -for security reasons- the operator does not open any other protocol, or because the terminal itself implements only the http protocol). Thus in order to be able to concentrate on their core activity, game developers are interested in middlewares dealing with networking for mobiles. These middleware functions are implemented mostly on the web server side, but also in the mobile phone.

Even though the standardization in the gaming community is not as advanced as CORBA or SOAP communities, this section presents the standards provided by the OMA. Then it studies the main proprietary solutions, before introducing GASP, an open source alternative.

1.1 OMA GAMING SERVICE PLATFORM PROPOSAL

First we look closely to the OMA proposal to define the services offered by a gaming service platform.

Open Mobile Alliance [OMA05] has been launched in 2002. 200 companies created OMA: Mobile network operators, mobile phone suppliers, software editors... Nokia, IBM, Motorola, Siemens, Intel, Sun for example belong to OMA. The goal of this standardization effort is to ensure a better interoperability and homogeneity between mobile technologies.

Among other activities, OMA includes a Game Services working group (OMA GS). Actually this group is the result of the integration of the Mobile Gaming Interoperability Group (MGIF) inside OMA. Created by Ericsson, Motorola, Nokia and Siemens, MGIF's goal was to create standard services for mobile based games.

The 1.0 specification [MGI02] is freely available. It is a Java API. It defines a reference architecture depicted in Figure 1. The following services should be provided by a fully conformant platform:

- Session Management
- Connectivity
- Measurements
- Traffic related events
- Game specific events
- Score and championship management
- Timers
- Log management

The OMA version 2.0 specifications [OMA04] extend the previous ones and are an ongoing work. New services deal with:

- Device capabilities
- Buddy list management
- Organization of competitions
- Messaging Services
- Localization
- Persistency
- Advanced Monitoring

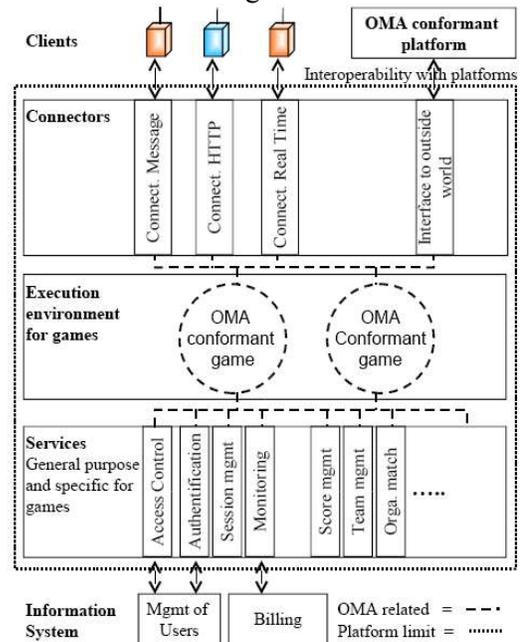


Figure 1: OMA/MGIF Game Service Architecture

The access to such services to create multiplayer games on mobile phones appears to be expensive. A small editing company can have some difficulties to buy them. When JIMM has been launched, we advocated that we had to support small game editors to promote multiplayer games on mobile phones. Small companies are creative, and efficient. We believe that an open source OMA conformant library could really help. GASP has been built for this purpose: It conforms the OMA GS version 1.0 and partially version 2.0 specifications.

1.2 STATE OF THE ART

Different platforms concurrent to OMA's specification exist. We will not present the whole list but only the ones that are relevant considering our project.

Several suppliers offer proprietary gaming service platforms for mobile games. Two approaches can be distinguished. The first one adapts an existing product to mobile game market. For example, Terraplay provides Terraplay-Move [TER05]. The second approach builds specific solutions for mobile games, Jive [PIX05] is an example of this approach.

SNAP Mobile is promoted by Nokia and Sun [NOK04]. It is firstly dedicated to community games, the N-GAGE being the first device targeted. SNAP Mobile is a global solution for wireless J2ME devices. The library is split into two parts: One on the mobile, the other one on the server. The client side interacts with the game logic ran on the server side. It seems that SNAP is proposed to telecom operators and to large game editor companies. Moreover SNAP seems to target Bluetooth® network games

The Exit Games Neutron [EXI05] platform is the only commercial platform implementing the OMA GS 1.0 specifications.

As far as we know, there is no open source gaming service platform for mobile games like GASP.

1.3 GASP DESIGN

1.3.1 Services implemented in the prototype

The MEGA project defined the services provided by our prototype. As we said previously the implemented services conform to OMA GS version 1.0 and partially 2.0 specifications in their current state [OMA04]. They rely on a database and a library that offers two levels of functionalities.

- Services for gamers are:
 - A lobby to identify on-going games and give the players the opportunity to meet,
 - Game management (join, create, start, stop, quit) and,
 - Account management (login, password, pseudo).
- System services consist in:
 - Session management,
 - Platform usage history,
 - Access control,
 - User authentication, and
 - Platform supervision.

The JIMM project stressed and enhanced these functionalities through the multiplayer game Kouiz.

1.3.2 UML model for GASP services

GASP conforms to all OMA communication interfaces that are linked to the game logic between the client and the server. Nevertheless the specified

data model (see figure 2) cannot be implemented directly. Thus we had to improve it. Doing so we clarified the function of each class and optimised the number of instantiated classes

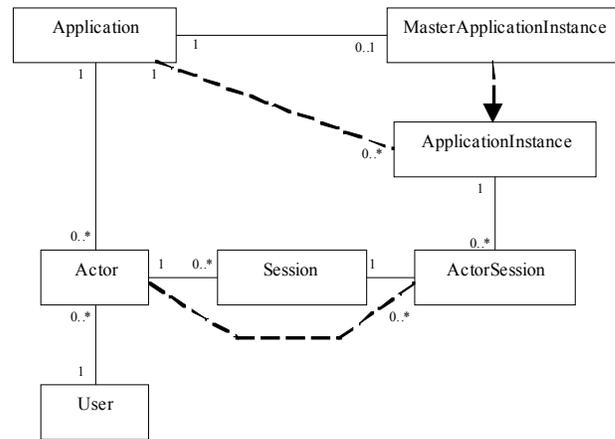


Figure 2: UML OMA data model

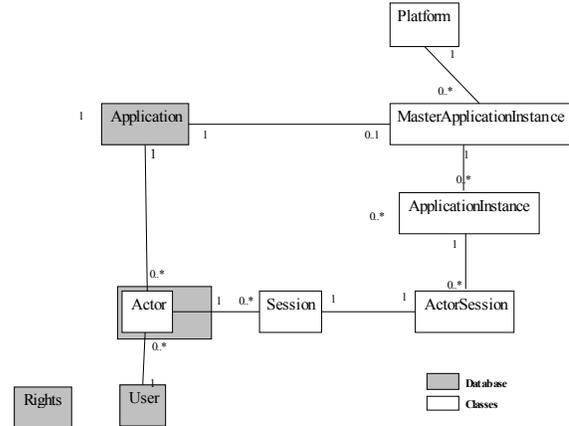


Figure 3: UML OMA data model enhanced due to GASP improvements

The following classes are finally used by the GASP middleware (see figure 3):

- **Platform [MGI02]:** This class represents the platform. It contains all the references of objects from GASP platform that are instances of classes defined hereafter. It is used as a container that allows to locate platform components. Location is more difficult with the OMA model proposal.
- **MasterApplicationInstance:** The role of this class has been made more explicit. It does not extend anymore ApplicationInstance class (as specified in the OMA model). It is dedicated to the management and maintenance of all ApplicationInstances of a given application.

- **ApplicationInstance [MGI02]:** Unchanged compared to initial standard. This class represents a game session of a particular application.
- **ActorSession [MGI02]:** Unchanged. This class represents a player connected to the game session.
- **Session [OMA04]:** Unchanged. This class represents a user connected to the platform.
- **Actor [MGI02]:** Unchanged, but each actor is instantiated using information stored in actors' database. This class represents the user profile for a specific application.

GASP library stores the following tables in a database:

- **Applications {AppID}:** Contains the AppID (ApplicationID) of applications available on the platform.
- **Users {UserID}:** Contains the UID (UserID) of users given access to the platform.
- **Actors {AID, UserID, AppID, login, pwd, lastPseudo}: Relation** user-application containing chosen login/password, as well as last used pseudo. It is identified by an AID (ActorID).
- **Rights {UserID, AppID}: Contains the pairs** (user, application) representing the access rights of users to the applications.

1.3.3 Services design

The previous section describes UML data model of services provided by the platform. This section briefly presents our design choices for GASP services. Several modules interact to provide the platform (see figure 4):

- **Platform Representation:** Set of class instances corresponding to the platform.
- **Servlet Container:** Container hosting GASP communication classes.
- **Game Servers:** Set of game logics supported by the server.
- **GASP DB:** GASP database.
- **Communication interfaces:** GASPClient and GASPServer are sets of classes to be extended by the game developer to take care of game communications.

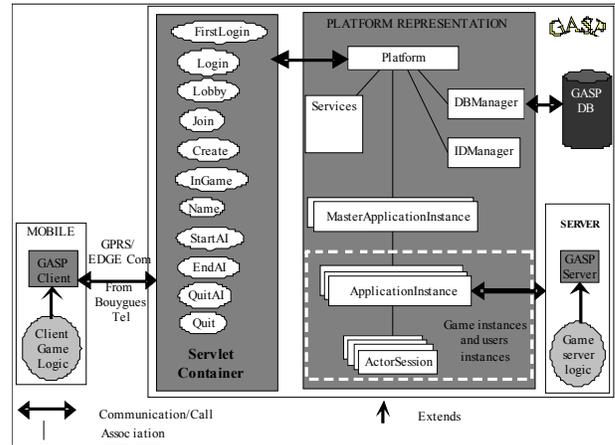


Figure 4: GASP design

At runtime (see figure 5), messages from mobile phones are processed by servlets at the server. These servlets interact with the instances hosted by the platform. This invokes the game server. Depending on the game logic (for instance, a dead reckoning algorithm may have been implemented to avoid too frequent update messages), messages are sent back to one or several mobiles.

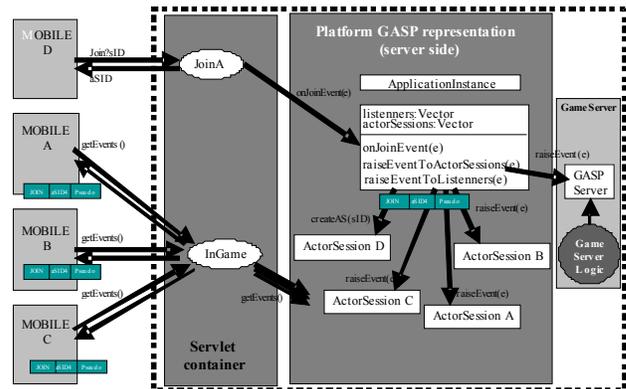


Figure 5: GASP event model

Notice communications between mobile phones and the server are based on an “HTTP over GPRS” protocol. The mobile sends messages as binary flows over http. Http responses “piggyback” messages from the server to the mobile. It is the only way to proceed, because http is the mandatory protocol supported on any mobile and operator’s network.

Message parameters are formatted according to the MoodS protocol. The MoodS protocol is a specific tool from GASP. In this protocol (see figure 6), game developer specifies, thanks to an XML file, the structure of messages. This XML file is then compiled to provide encoding/decoding classes

integrated to communication interfaces. Doing so, an optimal size of the corresponding classes is guaranteed, an optimal packet size (mandatory requirement to lower costs) and a full control on their memory allocation schemes. All these properties cannot be reached using standard communication libraries, like SOAP for instance.

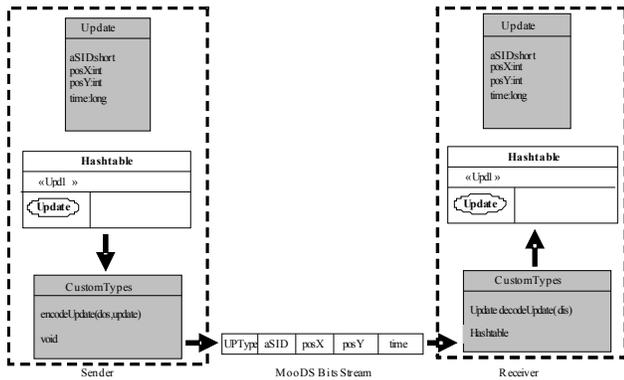


Figure 6: MoodS protocol

1.3.4 GASP implementation statements

GASP v1.0 is today operational, GASP client being available for MIDP and DOJA J2ME profiles. GASP is hosted by the ObjectWeb consortium as an open source software [OBJ05].

We now present the game Kouiz that relies on GASP.

2. KOUIZ

GASP is part of the JIMM project. Another part of it is Kouiz (Note that Kouiz should be published under another name unknown at the time of writing this article), a multiplayer game used as a proof of concept for GASP. Kouiz is being developed like a true multiplayer game for mobile phones and is being experimented over Bouygues Telecom mobile network. Kouiz in its current state is a java game intended to be downloadable at the end of the JIMM project.

2.1 KOUIZ GAMEDESIGN PRINCIPLES

Kouizs are animals and the principal characters of the game. Players are farmers and breed kouizs. A farmer, can create his own lineage of kouizs.

Kouiz can be seen as a two part game: Kouizfarm and Kouizmarket. The game can be played alone in the Kouizfarm universe. It can be played in a multiplayer mode in the context of Kouizmarket. Currently four players can play together at the same time in the virtual world of Kouizmarket.

The current version of Kouiz is implemented according to the DOJA 1.5 and DOJA 2.5 profiles. It

fully uses the gaming services provided by GASP: server and client libraries. The game is available, and a demo (that accelerates the game itself) can be presented.

Kouizfarm corresponds to the breeding place for kouizs. Some features are inspired by the Tamagotchi. Multiplayer phases enrich the gameplay: kouiz can be bought or sale at the market. Also, kouiz can fight during organized contests.

2.2 KOUIZ FARM

According to the game design, kouizs have been created by crazy biologists. They are vegetarians. They have different sizes, shapes, colours, abilities and moods. They have some basic intelligence. They evolve to improve their skills and to become a better challenger in the arena of multiplayer places. Kouizs grow in farms. Farmers breed kouizs and bring them to contests in order to be known and to earn money when their kouizs win.

When the player is alone, the gameplay, is related to feed, heal, train kouizs, and to garden for kouiz food. In the solo mode, the player learns kouiz's secrets, and expects to provide a specific lineage of kouizs.

Kouiz world is hardly policed. There are some basic rules that rely on plainly identified entities:

- The farmer is the player avatar. From a graphical point of view, all farmers have the same look (there is no personalisation currently): a farmer is named by the player to be distinguished from others. A farmer can carry only one kouiz at a time.
- Kouizs are the principal characters of the game. They grow up from a seed. Their characteristics are hard coded in their double DNA:
 - They have different representations: Triangle, square, rectangle, circle... and for a specific representation, the shape also varies. For example the rectangle represents a kouiz that can be thin or thick, tall or small. Form is related to mood. For example, a rectangle kouiz is angry or even aggressive (which can be useful during a fight in a competitive mode).
 - They have different colors. The color of a kouiz stays the same all along its lifetime but its brightness can change. This reflects its health. A pale kouiz is ill. Notice a kouiz never dies: When it is too badly ill, it becomes a seed, it is able to born again two days later. At this point it can be healed and energized.

- Kouizs have a mouth, two eyes and eyebrows. Eyes have their own characteristics: Color, form, place... Eyebrows depend on kouiz mood and mind. The mouth varies similarly to eyes.

Some kouiz' characteristics never change but others can be modified due to training or eating special food. Also a kouiz moves. The way it moves depends upon its representation. A triangle is able to jump for example. Finally, life points are associated to each kouiz, and are used during fights. They are also used when a kouiz is hungry, ill, tired or in a death state (The kouiz changes to a seed temporarily).

Kouizs interact with other kouizs. Especially, a kouiz borns when two kouizs (parents) are together. The baby kouiz inherits DNA from its parents. Its skills are randomly chosen from parents skills. Birth is allowed only in farms.

- Objects of the game area are: Trees, fences, buildings, containers, signs. They have special influences on food.
- Vegetables are kouizs' food. A kouiz eats when the farmer throws a vegetable towards it or when it is thrown towards a vegetable. Gardening of vegetables is of primary importance to feed kouiz. A vegetable has life points: When its reaches zero, it dies. Its effect depends upon its state. Vegetables have properties: toxic, regular food, sedative, stimulating, irritating, tonic. The effect of a vegetable depends on kouiz's state. Vegetables grow naturally or as a result of gardening. The time taken to grow is linked to the time of Kouiz world.
- Time in Kouiz world depends on the player time. In some way, Kouiz is real time. This real time is exactly handled for kouiz age and time spent in the sick room when they are ealed. The real-time life of kouiz is computed for some characteristics like mood and hunger. Real time is handled between two game sessions. When the player comes back to the game he has some work to do. This work is related to kouizs breeding and to gardening for kouiz's food. Time is fully used in the game design to enrich the game. Surprises can occur when the player goes back to his farm.

2.3 KOUIZ MARKET

Players alternate regularly solo phases and multiplayer phases. A market is a place where players meet together and where kouizs can participate to

contests (see figure 7 and 8): Fights. Kouiz is not a massively online multiplayer game: In the same market, at the same time, only four players can interact together. But, through web sites of players, best kouizs and so on, and through gamer communities, a wide range of gamers and their animals is expected to attend market places. This feature shall bring some massive aspect to the game. A market also allows to buy goods for breeding The farmer needs money. He earns money as a counterpart of throwing vegetables and kouizs inside its container. He can also buy a kouiz. In this case, the farmer should enter the market without a kouiz.



Figure 7: Farmers bring their kouiz to the market

The most interesting function of the multiplayer area is fight. To fight, two kouizs enter the ring.

- Each challenge is run in three rounds. The player chooses the three actions simultaneously before the challenge starts. When both players have chosen their actions, the game logic computes the corresponding points of damage. Damage points are compared to life points. The winner is the kouiz with the greatest number of life points. If life points become negative, kouiz dies (a seed).
- The behaviour of the kouiz is not deterministic. It depends on farmer orders (knock, jump, catch, slide), and, on the kouiz itself (DNA, training, mood). We will not go into details on how the right behaviour is chosen in this paper.
- The winner of a contest wins money, bonus, skills for future contests on the multiplayer world... It depends on the conditions of the contest: Bet, championship, challenge.



Figure 8: Kouizmarket, an example of fight

2.4 KOUIZ over GASP

Kouiz is implemented for Doja 2.5 and 1.5 mobiles. It fully uses GASP services. Currently, all the features described above have been implemented. The game is planned to be published for the last quarter of 2005.

The code provided by Kouiz Farm and Kouiz Market inside the player terminal corresponds for each application to about 30×Kb of code and 100×Kb of data when the game is running in a steady state.

The communication routines in Kouiz Market including GASP methods represent about 5Kb of code. Interactions during multiplayer phases generate around 150 bytes (rounded to 1 Kb by the operator for [HTTP request](#) billing) every 3 seconds.

GASP helped game programmers. They have been freed from the detailed implementation of interactions between mobile phones and the server that runs the game logic. MoodS especially eases the implementation of exchanges through XML specifications.

Finally, we have evaluated the feasibility of time constrained games over GPRS network. Throughout development of Kouizmarket, we were confronted with the high latency of this type of network, about one second and half. Real time games require a latency lower than 200ms to work under ideal conditions in the majority of multiplayer game types, excluding Real Time Strategy games for which 500ms is bearable [SME01]. Kouizmarket is definitely not a turn by turn game: It may be considered as a pseudo real time game. Indeed Kouizmarket looks as a real time game on mobiles which are not side by side (which shall be the case under real gaming sessions). This has been made possible thanks to a specific dead reckoning algorithm and the game design which avoids the management of collisions with other players.

3. CONCLUSION AND FUTURE WORKS

3.1 Current Statements

This paper presents GASP, a gaming services platform for multiplayer mobile game, which conforms to OMA specifications, and a multiplayer game that fully uses GASP features. Our experiment allowed improving the OMA specification proposal. The implementation showed the need to modify the class model of application and session instances.

Moreover, Kouiz is a concurrent game in real time. It can be qualified as a pseudo real-time game. It differs from turn by turn games considered by OMA in their specification. Then our experience of Kouiz could help to enhance OMA gaming services specifications.

In the context of the JIMM project, GASP is published through the ObjectWeb consortium as an open source middleware in order to give editing companies the ability to make trials of multiplayer mobile games. In parallel we improve the GASP library to facilitate the publishing and the deployment of several games with different characteristics.

3.2 Enhancements

To go further, GASP needs some enhancements. We could think about extensions like championship or competition between players management. But more important features have to be designed.

First, game life cycle management has to be implemented. Currently in GASP there is two types of application life cycle implemented: a manual mode and an automatic mode.

- In manual mode the game session has an owner responsible for its creation. The owner controls the life cycle of the game, i.e. he starts, ends or destroys the game session.
- There is a game room approach of the game session: Players are waiting for sufficient number of players then the owner can start the game. This mode is the mode introduced in the OMA specifications.

But this mode is also not sufficient as each application can require a specific life cycle. For example in the context of Kouizmarket we have introduced an additional mode, the automatic mode because the Kouizmarket game session has no owner: It starts when a player arrives and it is destroyed when the last player leaves the game session.

We need a generic service to describe the game life cycle. We have to avoid this tricky business for game programmers. We have to provide a comprehensive model, an API and an efficient library.

Next, GASP has to provide persistency. To address online games this feature is mandatory. In the context of games for mobile phone this need is enforced. We have to take into account the fact that communications can be cut; game can be interrupted by a phone call... In the context of Kouiz, the state of the game is stored because the player can change from solo mode to multiplayer mode and vice-versa.

In Kouiz Market persistency is hard coded in server game logic. During the multiplayer session, the server logic is connected to a specific database for the game and stores the critical informations of player kouiz dna, sold or statistics after a kouiz sale, a kouiz purchase or after a fight.

That type of implementation may cause problems with database connections management in game server logics, thus GASP has to provide a generic service for game persistency.

We have identified a final high-priority requirement: Interfaces with outside world information systems have to be offered. Such an interface is specific today. In the future the game services will have to communicate with the game editor company and the network operator information systems (player profile, game updates, accounting, bonus management, events, data mining...). For example, this kind of facility can be useful to build data mining tools and can help to determine the behavior of players during the game.

4. REFERENCES

[EXI05] EXIT-GAMES, http://www.exit-games.com/BaseCMP/pages/contentdelivery?CID=9429&JOB_NAME=DISPLAY_PAGE

[MGI02] MGIF. *MGIF platform specification v1.0*, http://www.openmobilealliance.org/release_program/g_s_v10.html, 2002

[NOK04] Nokia. SNAP Mobile. Mobile Online Game and Java Community Platform. http://snapmobile.nokia.com/n-gage/web/en/snapmobile/snapmobile_home.jsp. Mars 2004.

[OBJ05] ObjectWeb. *ObjectWeb Forge: Project Info-GASP*, <http://gasp.objectweb.org>, June 2005.

[OMA04] Open Mobile Alliance. *Gaming Platform Version 2.0: Client/Server Protocol* (Draft version 0.8.1), March 2004

[OMA05] Open Mobile Alliance. *OMA home*, <http://www.openmobilealliance.org/>, 2005

[PIX05] Pixel technologies – ITV, Mobile and PC Entertainment & Edutainment Channels, <http://www.pixel-tech.com/mobile.html>, 2005

[SCH02] C.Schaefer, T.Enderes, H.Ritter, M.Zitterbart, *Subjective Quality Assessment of Multiplayer Real Time Game*, Proceedings of The International Workshop on Network and System Support for Games NetGames, 2002, pp. 9-12.

[SIM04a] M. Simatic, S. Craipeau, A. Beugnard, S. Chabridon, MC. Legout, E. Gressier-Soudan. *Technical and Usage Issues for Mobile Multiplayer Games*. In 5th International Conference on Computer Games: Artificial Intelligence, Design and Education, CGAIDE, 2004.

[SIM04b] M. Simatic. *Mobile multiplayer online Games Architecture (MEGA)*. http://bscw.enst-bretagne.fr/pub/bscw.cgi/d3070714-1*/MEGA_Presentation_VE.html. April 2005

[SME01] Jouni Smed, Timo Kaukoranta, Harri Hakonen, *Aspect of Networking in Multiplayer Computer Games*, Proceedings of The International Conference on Application and Development of Computer Games in the 21st Century, Nov. 2001. pp. 2, B. *Latency*.

[TER05] Terraplay. *Terraplay Move: mobile multiplayer gaming service*, <http://www.terraplay.com>, 2005

