

GASP : Un intergiciel pour les jeux en réseaux multijoueurs sur téléphones mobiles

R. Pellerin
CNAM-CEDRIC
292, rue St Martin
75141 Paris Cedex 03
Tel : 33 1 58 80 85 23
France
pellerin@cnam.fr

F. Delpiano
Pastagames
57, rue St Anne
75002 Paris
Tel : 33 1 58 62 53 00
France
fabien@pastaga.net

E. Gressier-Soudan
CNAM-CEDRIC
292, rue St Martin
75141 Paris Cedex 03
Tel : 33 1 58 80 85 23
France
gressier@cnam.fr

M. Simatic
GET-INT
9 rue Charles Fourier
91011 Evry Cédex
Tel : 33 1 60 76 45 61.
France
michel.simatic@int-evry.fr

RÉSUMÉ

Ce papier décrit des travaux en cours concernant la réalisation d'une bibliothèque de services pour des jeux multijoueurs en réseau sur téléphone mobile. Cette bibliothèque, appelée GASP, est conforme aux spécifications v2.0 de l'*Open Mobile Alliance* (OMA). Elle est programmée en Java et est constituée d'une partie serveur, et d'une partie cliente sur téléphone mobile pour les deux profils J2ME que sont MIDP 2 et DOJA 1.5. GASP v1.0 implante partiellement la spécification OMA : nous travaillons sur une nouvelle version plus complète.

Mots Clefs

Jeu, multijoueur, téléphones mobiles, OMA, MIDP, DOJA

ABSTRACT

This paper describes an ongoing work. We are designing and implementing a library of services dedicated to multiplayer on-line games for mobile phones. This library, called GASP, conforms to *Open Mobile Alliance* (OMA) v2.0 specifications. It is implemented in Java and offers a server and a client part. The latter is defined for MIDP 2.0 and DOJA 1.5 J2ME profiles. GASP v1.0 implements partially OMA specification: We are working on a new version more complete.

Categories and Subject Descriptors

K.8.0 [General]: Games, C.2.1 [Network Architecture and Design]: Wireless Communication

General Terms

Design, Standardization

Keywords

Game, multiplayer, mobile phones, OMA, MIDP, DOJA

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

UBIMOB 05, May 31. - June 3, 2005, Grenoble, France.

Copyright 2005 ACM X-XXXXX-XXX-X/XX/XXXX \$5.00.

1. INTRODUCTION

Le travail décrit dans ce papier est issu du projet MEGA, financé par le Groupement des Écoles des Télécommunications, qui s'est terminé en décembre 2004. Ses objectifs étaient [3] :

- **Évaluer les aspects techniques liés aux jeux multijoueurs sur mobiles.** Un certain nombre de verrous technologiques et scientifiques qui limitent les jeux sur mobiles d'aujourd'hui et de demain ont été identifiés et caractérisés. Certains des points durs notoires tels que : intergiciels pour les jeux sur mobile, abstractions de communication, prise en compte des temps de latence, algorithmes de cohérence, bases de données ont été étudiés dans ce cadre. Dans le contexte de MEGA la plateforme GASP, présentée dans ce papier, a servi de prototype démontrant la faisabilité des jeux multijoueurs en réseau sur mobiles.
- **Étudier les aspects psychologiques et sociologiques exprimés par les consommateurs/joueurs.** Il paraissait nécessaire d'articuler pratiques de mobilité, pratiques ludiques, et usage de portables. Aussi, les usages potentiels des jeux interactifs en situation de mobilité ont été identifiés, en étudiant les pratiques quotidiennes de mobilité, la façon dont les joueurs utilisateurs de portables investissent l'espace et le temps, ainsi que la façon dont ils investissent leurs engagements sociaux.

Les résultats techniques, scientifiques et sociologiques ont été présentés dans [4]. Les points les plus significatifs que nous avons retenus pour l'élaboration de jeux multijoueurs sur mobiles sont :

- L'utilisateur de mobile est fondamentalement nomade, à la différence d'un PC ou d'une console.
- Le mobile est un objet très personnel multi-fonctions (téléphonie, messages, réveil, jeux...) qui accompagne l'utilisateur où qu'il soit : dans les transports, en classe, chez lui, au travail, dans son sommeil.
- Le mobile est un objet de mise en relation : il a une vocation sociale. Il peut permettre d'être en interaction avec le monde extérieur, l'interaction étant dans les deux sens.
- Comme tout équipement embarqué il a des possibilités limitées en terme de puissance et surtout en terme d'IHM.

- Les communications posent des contraintes technologiques dont on ne peut s'affranchir : grande latence, gigue, débit faible, taux de perte, taux d'erreur et taux de déconnexion importants.

Par rapport à ces différents aspects, la mise en relation et la gestion de joueurs en mobilité nous a semblé la première brique de base d'un intergiciel de jeux en réseau. Ce papier décrit cette brique, dénommée GASP. Après un état de l'art des intergiciels concurrents dans le paragraphe 2, le papier poursuit dans le paragraphe 3 par la présentation de son architecture et d'un premier prototype. Le paragraphe 4 conclut, et présente les perspectives du travail en cours.

2. ÉTAT DE L'ART

Un état de l'art complet peut être trouvé dans [2]. Nous reprenons ici les travaux qui nous ont semblé les plus significatifs.

Plusieurs fournisseurs ont senti la nécessité d'offrir une architecture de services pour les jeux sur mobiles. On identifie deux approches : la première est l'adaptation d'une plateforme existante pour les jeux multijoueurs en environnement fixe (typiquement Terraplay se décline en Terraplay-Move [7]) ; la seconde consiste à construire une solution spécifique pour les jeux sur mobiles (typiquement Jive [8]).

SNAP Mobile est un projet proposé par Nokia et Sun [6]. Il offre une bibliothèque de services et des outils de développement pour la réalisation de jeux en communauté ou en ligne multijoueurs sur des terminaux mobiles. La console de jeux/téléphone mobile N-GAGE est un représentant type des terminaux visés par cette offre. SNAP Mobile est une solution globale pour des terminaux J2ME sans fil qui consiste en une partie cliente sur le mobile qui interopère avec la logique de jeux sur un serveur. Cette offre est orientée opérateur télécom, ou gros éditeurs. En l'état, et compte tenu de ce que nous avons pu lire dans la presse, SNAP Mobile sur Nokia vise les jeux en réseau (notamment Bluetooth®), l'accent étant mis sur la fourniture de fonctionnalités communautaires de joueurs. Il est difficile d'avoir des informations précises, mais il semble que ses objectifs soient proches de ceux de GASP.

À notre connaissance, il n'existe pas de projet logiciel libre concurrent de GASP.

L'*Open Mobile Alliance* (OMA) [9] a été formée en 2002 par un groupement de plus de 200 entreprises d'horizons divers : opérateurs mobile, entreprises de fabrication de terminaux mobiles, entreprises spécialisées dans les réseaux, dans les technologies de l'information et fournisseurs de contenu. On retrouve au sein de l'OMA : Nokia, IBM, Motorola, Siemens, Intel, Sun... Cet effort de standardisation s'inscrit dans une volonté d'homogénéisation et d'interopérabilité des technologies pour mobiles.

Le Mobile Gaming Interoperability Group (MGIF) qui a rejoint l'OMA a été créé par Ericsson, Motorola, Nokia et Siemens. Le MGIF avait pour objectif de créer des services standards pour les jeux. Il a donné lieu à la création du groupe de travail « *Games services* » au sein de l'OMA. L'architecture d'une plateforme conforme à OMA/MGIF a deux objectifs prioritaires : l'interopérabilité afin de partager des ressources ou des services, et la portabilité. Les services proposés par l'API OMA/MGIF sont interfaçables avec des services spécifiques à l'opérateur tels que

les services de facturation ou encore les services d'authentification utilisateur.

La spécification en version 1.0 [5] est disponible librement sous forme d'API Java et offre, dans le cadre d'un modèle d'architecture (cf. figure 1), les services ci-dessous :

- **Gestion de session** : c'est le cœur de la plateforme établissant le lien entre l'utilisateur et l'instance du jeu côté serveur. Cette API contrôle le cycle de vie de l'application. Les entités principales constituant l'instance d'une session sont les instances des classes ActorSession, représentant l'utilisateur, et ApplicationInstance, représentant l'application.

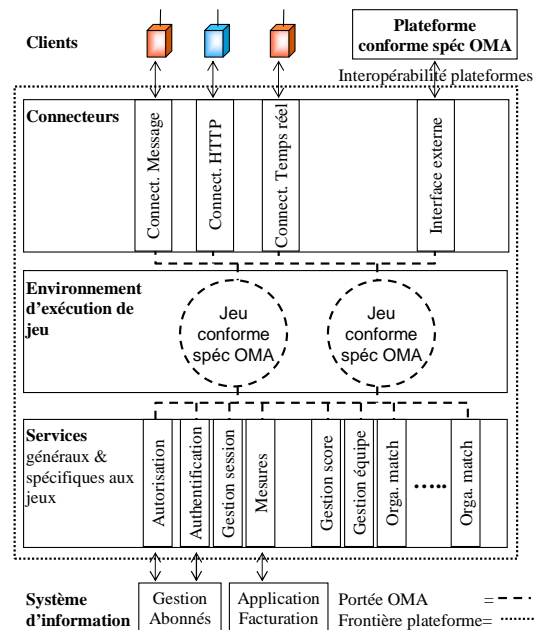


Figure 1. Architecture Game Service de l'OMA/MGIF

- **Connectivité** : service qui permet d'établir la communication entre les clients et l'application, comment les requêtes des clients vont être acheminées et comment les réponses de l'application vont être générées et retournées aux clients. Le modèle de communication est caractérisé par 4 méthodes : *client pull*, *client push*, *application pull* et *application push*. Les modes de communications pris en compte sont : asynchrone et synchrone. À noter que, dans cette version, seuls les clients de type messagerie ou navigateur sont adressés, les clients exécutables seront intégrés dans les futures versions de l'API.
- **Mesure** : ce service permet de faire remonter des événements pour leur utilisation dans la facturation, ces événements peuvent être de type trafic ou spécifique au jeu.
- **Événements de type trafic** : durée de la session, taille des données transférées, autorisations de souscription/pré-paiement.
- **Événements spécifiques au jeu** : ce type d'événements (par exemple le changement de niveau) est créé dans l'instance de la classe ActorSession. Les événements sont paramétrables.
- **Gestion des scores et des compétitions** : cette API fournit les mécanismes d'enregistrement et de récupération des scores. Cette API permet également la mise en place de nombreuses formes de compétitions.

- **Timers** : les applications nécessitant une gestion de temporisations font appel à cette API. Ces Timers sont présents au sein des classes `ApplicationInstance` et `ActorSession`. Des timers spécifiques pour les fichiers de configuration ou les outils d'administration peuvent être créés via la classe `MasterApplicationInstance`.
- **Logging** : service permettant la création de logs pour le débogage, le repérage d'erreurs et le monitoring via l'interface `Logger`.

Face aux nombreux services fournis par les plates-formes propriétaires vues précédemment, les fonctionnalités spécifiées dans MGIF v1.0 peuvent paraître réduites. La standardisation des services est un passage obligé pour éviter un dédale de technologies propriétaires offrant les mêmes fonctionnalités.

La plateforme version 2.0 est en cours de spécification à l'OMA. Elle fournit les services supplémentaires suivants [10] :

- Capacités du terminal
- Gestion « buddy », réseau d'amis
- Organisation de compétitions
- Informations via messagerie
- Localisation
- Persistance des données
- Monitoring avancé

Pour un éditeur de jeux multijoueurs, intéressé par l'ensemble de ces services, l'accès à une plateforme de services apparaît comme un investissement non seulement technique, mais aussi financier. Cet effort ne peut être consenti par tous, en particulier les petits éditeurs. La créativité et le dynamisme dans la création doivent être favorisés quelle qu'en soit l'origine. Il nous a donc semblé indispensable d'offrir une bibliothèque de base de services pour les jeux multijoueurs sur mobiles en logiciel libre pour aider les petits éditeurs à accéder à ce marché des jeux en réseaux sur mobiles, ceci afin de sortir de l'ornière des jeux assez pauvres qui n'ont rien à envier aux jeux sur console en 2D vieux d'une dizaine d'années. Le choix d'un standard et donc de la spécification de l'OMA était alors naturel. GASP est une implantation partielle de cette spécification.

3. GASP : UNE MISE EN ŒUVRE DU MODELE DE L'OMA

3.1 Les Services mis en œuvre dans le prototype

L'ensemble des services offerts par le prototype issu du projet MEGA est une première étape. Les services mis en œuvre sont conformes aux spécifications v2.0 de l'OMA, et s'appuient sur une base de données et une bibliothèque qui se découpe en deux parties. Les services clients sont constitués : du lobby pour la localisation de parties et la rencontre de joueurs, la gestion de jeu (rejoindre, créer, démarrer, stopper, quitter une partie) et la gestion de compte (login, mot de passe, pseudonyme). Les services systèmes sont constitués : de la gestion de session, de l'historique d'utilisation de la plateforme, du contrôle d'accès, de l'authentification de l'utilisateur par login et mot de passe, et de la surveillance de la plateforme.

3.2 Modèle UML des Services

GASP respecte les interfaces de communication de la spécification OMA relatives à la logique du jeu entre client et serveur. Toutefois, le modèle proposé par la norme n'était pas implantable en l'état. Un modèle amélioré qui ne touche que le fonctionnement interne de la plateforme a été fourni. L'ensemble des améliorations apportées au modèle de Gaming Service Platform de l'OMA, v2.0, et constituant le modèle de GASP permet une clarification des rôles de chaque classe et une optimisation du nombre de classes instanciées.

La bibliothèque GASP utilise les classes suivantes :

- **Platform [5]** : c'est la classe représentant la plateforme, c'est-à-dire contenant toutes les références aux objets de la plateforme GASP qui sont des instances des classes définies ci-dessous. Cette classe sert donc de conteneur et permet ainsi de localiser les éléments constituant la plateforme, localisation qui semble difficile dans le modèle de l'OMA.
- **ApplicationInstance [5]** : rôle non modifié par rapport à la proposition de la norme initiale.
- **MasterApplicationInstance** : le rôle de cette classe a été modifié ou plutôt précisé. Elle ne dérive plus de la classe `ApplicationInstance` comme dans le modèle de l'OMA. Elle représente la classe chargée de gérer et de maintenir l'ensemble des `ApplicationInstances` d'une application donnée.
- **Actor [5]** : rôle non modifié, mais chaque acteur est instancié à partir des informations contenues dans la base de données des acteurs.
- **Session [10]** : rôle non modifié.
- **ActorSession [5]** : rôle non modifié.

La bibliothèque GASP définit en Base de Données les tables suivantes :

- **Applications {AppID}** : contient les AppID (`ApplicationID`) des applications disponibles sur la plateforme.
- **Users {UserID}** : contient les UID (`UserID`) des utilisateurs ayant accès à la plateforme.
- **Actors {AID, UserID, AppID, login, pwd, lastPseudo}** : relation utilisateur-application, contenant les login et mot de passe choisis, ainsi que le dernier pseudonyme utilisé, identifiée par un AID (`ActorID`).
- **Rights {UserID, AppID}** : contient les doublons (utilisateur, application) représentant les droits d'accès des utilisateurs aux applications.

3.3 Architecture de la bibliothèque de services

La section précédente décrit le modèle UML des services supportés par la plateforme. Ce paragraphe présente brièvement les choix architecturaux relatifs aux communications pour mobiles avec GPRS. Ces requêtes pour GASP v1.0 utilisent le protocole "HTTP over GPRS". L'architecture de GASP se compose de plusieurs modules qui interagissent :

- **Platform Representation** : ensemble des instances des classes constituant la plateforme.
- **Servlet Container** : conteneur accueillant les classes de communication de GASP.

- **Game Servers** : ensemble des logiques de jeu serveur supportées en exécution.
- **Interfaces de communications** : GASPCClient et GASPServer.
- **GASP DB** : base de données de GASP.

Pour plus de détails sur la mise en œuvre, nous renvoyons le lecteur à [1]. L'architecture est résumée dans la figure 2.

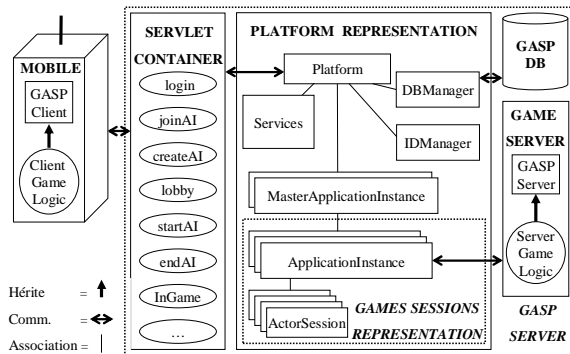


Figure 2. Architecture de la bibliothèque de Services GASP

3.4 État d'avancement de GASP v1.0

Dans sa première version, GASP v1.0 propose un nombre de services restreint :

- **Services clients (GASPCClient)** : le lobby pour la localisation de parties, la gestion de jeu (rejoindre, créer, démarrer, stopper, quitter une ApplicationInstance) et la gestion de compte (login, mot de passe, pseudonyme).
- **Services systèmes (GASPServer)** : la gestion de session, l'historique d'utilisation de la plateforme (*logging*), le contrôle d'accès grâce à la gestion de la table des droits au niveau de la base de données de GASP, l'authentification de l'utilisateur par login et mot de passe et le monitoring de l'état de la plateforme au moment de l'envoi de la requête au servlet PlatformState retournant un flux html.

L'ensemble s'exécute sur des terminaux J2ME suivant les profils MIDP 2 et Doja 1.5. Le protocole d'échange de données a été optimisé afin de minimiser la taille des messages et donc réduire le temps des communications et la taille des données transférées entre la logique client et la logique serveur.

Pour faire un bilan du prototypage, un jeu qui utilise GASP est mis en œuvre. L'utilisation de mobiles comme terminaux de jeu nous confronte aux limites d'IHM, puissance de calcul... de ces machines [11]. De plus, le réseau mobile a pour conséquence des temps de latence fort dont il faut tenir compte au niveau du *game design*. Les problèmes rencontrés sont majoritairement des problèmes de portage liés aux différents terminaux mais surtout au modèle de programmation pour les deux profils DOJA et MIDP : la gestion des données rémanente est complètement différente (*scratchpad vs RMS*), et la taille mémoire disponible pour les programmes n'a pas les mêmes limites (DOJA impose un maximum de 30 Ko pour les JAR). Par ailleurs, les communications doivent s'appuyer sur le protocole http (seul protocole disponible sur tous les mobiles) qui ne permet pas d'implanter naturellement un modèle d'interaction producteur-consommateur, il faut en fait détourner le modèle client-serveur en laissant pendante une requête du client. Enfin, il faut

compacter le volume des données transférées : la facturation sur les réseaux GPRS s'effectue au volume, et la taille des messages est limitée.

4. CONCLUSION ET PERSPECTIVES

À ce jour, GASP existe à l'état de prototype montrant sa faisabilité. Il a permis d'évaluer la spécification OMA [10] qui, en l'état, n'est pas suffisante pour développer un jeu multijoueur, et il a fallu par exemple modifier le modèle de classe des instances d'application et des instances de session. Plus d'informations sont disponibles dans [1].

Du point de vue des perspectives, dans un premier temps, il faut consolider la partie technique : c'est l'objectif du projet JIMM labellisé RIAM débuté depuis octobre 2004 avec Bouygues Telecom et Pastagames un éditeur de jeux sur mobiles. L'objectif est d'enrichir la bibliothèque GASP pour faciliter la publication et le déploiement de plusieurs jeux avec des caractéristiques différentes (le *framework OSGi* [12] est une piste à évaluer pour répondre à certains de nos besoins).

Dans un second temps, une fois que cette brique de base sera opérationnelle, nous pourrons tirer parti des caractéristiques spécifiques des mobiles, en particulier de la face sociale attachée aux mobiles. Les futurs services de GASP devront permettre la conception de jeux qui intègrent les nouvelles fonctions des mobiles et leurs accessoires : vidéostreaming, géo-localisation, équipements Bluetooth@... et qui immergent le joueur dans un univers où virtuel et réel se superposent et se complètent.

5. REFERENCES

- [1] R. Pellerin. *Gaming Services Platform, Plateforme pour les jeux multijoueurs sur mobiles*. Mémoire de DEA Cnam/ENST/Paris 6. Septembre 2004
- [2] R. Pellerin. *Mobile Gaming Services Services pour les jeux multijoueurs sur mobiles*. Rapport bibliographique DEA Cnam/ENST/Paris 6. Juin 2004.
- [3] M. Simatic, S. Craipeau, A. Beugnard, S. Chabridon, MC. Legout, E. Gressier-Soudan. *Technical and Usage Issues for Mobile Multiplayer Games*. 5th CGAIDE. Reading. England. November 2004.
- [4] *Architecture pour jeux multijoueurs en ligne sur mobile (MEGA)*. http://bscw.enst-bretagne.fr/pub/bscw.cgi/d3043906-1/*MEGA_Presentation_VF.html. Mars 2005
- [5] MGIF. *MGIF platform specification v1.0*, http://www.openmobilealliance.org/release_program/gv10.html, 2002
- [6] Nokia. SNAP Mobile. Mobile Online Game and Java Community Platform. http://snapmobile.nokia.com/n-gage/web/en/snapmobile/snapmobile_home.jsp. Mars 2004.
- [7] Terraplay. *Terraplay Move : mobile multiplayer gaming service*, <http://www.terraplay.com>, 2003
- [8] Pixel technologies – ITV, Mobile and PC Entertainment & Edutainment Channels, <http://www.pixel-tech.com/mobile.html>
- [9] OMA home, <http://www.openmobilealliance.org/>
- [10] Open Mobile Alliance. *Gaming Platform Version 2.0: Client/Server Protocol (Draft version 0.8.1)*. Mars 2004
- [11] Nokia. *Introduction to Mobile Game Development (Version 1.1)*, <http://www.forum.nokia.com/main.html>, Janvier 2003
- [12] OSGi home, <http://www.osgi.org>