

Vers des jeux sérieux massivement participatifs permettant une co-conception des règles par les acteurs

Nour El Mawas, Jean-Pierre Cahier, Aurélien Bénel

▶ **To cite this version:**

Nour El Mawas, Jean-Pierre Cahier, Aurélien Bénel. Vers des jeux sérieux massivement participatifs permettant une co-conception des règles par les acteurs. H2PTM'11 : Hypermédias et pratiques numériques, Oct 2011, Metz, France. pp.79-92. hal-03259992

HAL Id: hal-03259992
https://hal.science/hal-03259992v1

Submitted on 19 Nov 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Vers des jeux sérieux massivement participatifs permettant une co-conception des règles par les acteurs

Nour El Mawas[1], Jean-Pierre Cahier[1], Aurélien Bénel[1],

[1] Laboratoire ICD/Tech-CICO
(Technologies pour la Coopération, l'Interaction et les Connaissances dans les Collectifs)
UTT (Université de Technologie de Troyes) 10010 Troyes,
{ nour.el_mawas , cahier , aurelien.benel }@utt.fr

Résumé : Cet article propose de définir des jeux sérieux « participatifs » c'est-à-dire favorisant la coopération à la fois dans la communauté des joueurs (dans un contexte massivement multi-joueurs) et dans la communauté des concepteurs du jeu. Pour cela nous étudions les conditions fonctionnelles et architecturales nécessaires pour organiser, contrôler, consulter, et éditer les informations concernant le jeu, tenant compte des impératifs de participation, de performance et de souplesse, lorsque les participants sont nombreux et pas nécessairement du même avis. Nous proposons des lignes directrices pour une infrastructure souple devant permettre non seulement de gérer le déplacement des joueurs ou les effets des actions accomplies sur une scène (y compris leurs effets sur les autres scènes) mais aussi d'assurer la conception participative des objets et des règles du jeu, impliquant d'en débattre entre les concepteurs. Pour cette infrastructure nous avons choisi d'utiliser un système de base de données NoSQL (CouchDB), et nous utilisons cette infrastructure pour illustrer certaines de ces idées, en prenant l'exemple du jeu SOS-21 pour l'apprentissage de bonnes pratiques en développement durable.

Mots-clés : jeu sérieux, coopération, forum de discussion, base de données NoSQL, multijoueurs, développement durable

1. Introduction

Ces réflexions s'inscrivent dans le cadre du développement de la version participative et multi-joueurs du jeu SOS-21[1], un « jeu d'aventure » en correspondance avec des territoires visant à sensibiliser les acteurs du grand public (étudiants, adolescents, habitants de ces territoires…) à l'impact de leurs actes quotidiens sur le développement durable. Dans notre approche, les objets du jeu et ses règles sont par nature très nombreux et diversifiés. Ils doivent pouvoir résulter d'une conception décentralisée et participative confrontant de nombreux concepteurs, pédagogues,

[1] http://sos-21.sourceforge.net/

partenaires locaux, experts ou scientifiques de diverses disciplines, qui connaissent et considèrent le jeu, les territoires, les problèmes et les évaluations de façons complémentaires ou croisées. Beaucoup de jeux sérieux apparaissent pour l'apprentissage de pratiques complexes -par exemple en matière de développement durable, de sécurité ou de gestion des risques en environnement urbain - où les connaissances de référence ne sont pas forcément stabilisées ni unanimes. Elles sont particulièrement dynamiques et en évolution constante. Cela amène à souhaiter que les règles, comme d'autres éléments du jeu, puissent facilement commentées, mises en discussion et modifiées par les acteurs eux-mêmes, sans avoir à repasser par les informaticiens.

La suite de cet article est découpée comme suit. La section 2 présente un approfondissement de notre approche sur la notion de jeu participatif. La section 3 illustre cette approche avec l'exemple du jeu SOS-21, en détaillant les formes nécessaire aux aspects participatifs dans le cas de ce jeu. La section 4 présente un aperçu de l'architecture utilisée et des solutions techniques adoptées pour supporter ces aspects participatifs, en particulier en termes de conscience mutuelle (asynchrone et synchrone) des participants, et de discussion / édition des règles du jeu. Enfin, la section 5 conclut cet article et présente les perspectives.

2. Les bases d'un jeu sérieux participatif

La notion de co-conception dans une approche participative remonte à des projets des années 80, en lien avec une réflexion sur la démocratie, dans des domaines comme la réparation de locomotives ou le monde de l'édition dans le monde scandinave (Ehn 02) ou la conception de systèmes d'information et de coopération (Winograd 1986). Ces auteurs soulignent la nécessité d'inclure très tôt toutes les personnes concernées dans la conception. Pour devenir « modélisation engagée » (Zacklad 2003), la co-conception doit en outre être accompagnée de la construction par les acteurs de structures tels que des « cartes » de leurs connaissances, en rapport avec leurs pratiques. Il devient ainsi possible pour les acteurs de modeler eux-mêmes l'organisation de leur espace de coopération ou de « laisser émerger » cette organisation au fil de l'eau.

L'espace du jeu sérieux « participatif » est donc pour nous un espace co-conçu et qui doit proposer des fonctions riches. Il doit permettre non seulement de *confronter massivement de multiples joueurs* utilisant les objets existants de cet espace, mais aussi que *l'espace du jeu puisse être facilement co-construit* par un groupe large, par ajout, modification et discussion de nouveaux objets, connaissances et règles. L'espace du jeu est donc participatif à la fois pour le groupe des joueurs et pour celui des concepteurs. Ces deux groupes ne sont pas d'ailleurs totalement cloisonnés : certains joueurs, par exemple les plus expérimentés ou créatifs, ayant l'idée d'améliorations ou de nouveaux services, pourraient être conviés à rejoindre le groupe des concepteurs ou tendent à peser sur lui pour obtenir des changements.

La possibilité « d'architectures de participation » ambitieuses pour les univers virtuels a été confortée par le succès d'applications comme Wikipedia ou plus généralement des applications Web 2.0 (O'Reilly, 2005). Mais réaliser ces architectures pose encore de nombreux problèmes, car il faut au niveau de l'infrastructure tenir compte du grand nombre de joueurs, introduire une certaine souplesse pour tenir compte des apports des acteurs multiples, que ceux-ci soient joueurs ou concepteurs. De plus,

les acteurs doivent croiser leurs compétences dans des situations pour lesquelles les connaissances et les données sont très nombreuses et fortement évolutives, ce qui est le cas dans les jeux ou les scènes et leurs items sont très nombreux.

La figure 1 ci-après illustre l'infrastructure et les aspects participatifs au niveau des deux communautés : co-concepteurs et joueurs. Au niveau de l''infrastructure, pour structurer le niveau de catalogage des items du jeu, représenté dans le plan inférieur de la figure, nous nous inscrivons dans la tradition des modèles de connaissances facilitant la co-conception coopérative, tels que Hypertopic (Zhou *et al.,* 2006), en cherchant à l'étendre au contexte particulier des jeux. Concernant les aspects participatifs, dans le plan intermédiaire, les co-concepteurs peuvent chercher, ajouter, cataloguer, discuter et commenter les items (les règles du jeu et les autres éléments de conception) comme la scène par exemple). De leur côté les joueurs peuvent chercher, commenter les items et les utiliser pour jouer.

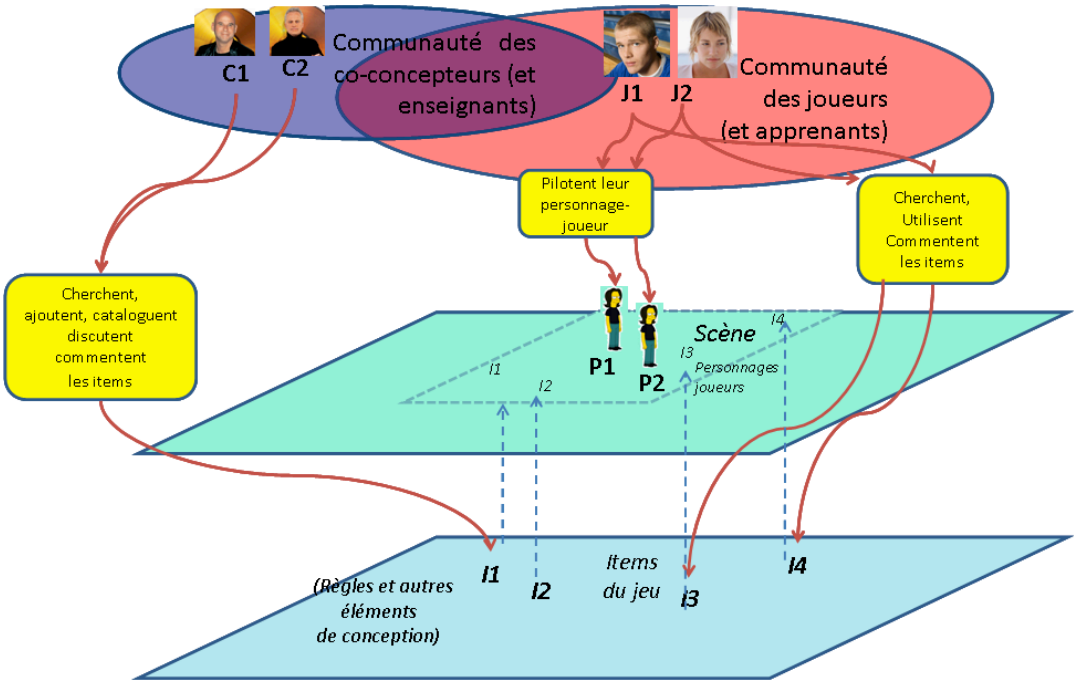


Figure 1. Co-conception dans les jeux sérieux massivement participatifs

Pour les concepteurs une discussion permanente, de type forum / commentaire de blog, doit pouvoir être organisée sur tout item du jeu en particulier sur chaque règle, y compris et en particulier quand le jeu est en service, car c'est à ce moment que les différences d'interprétation, d'opinions et d'approches sont les plus faciles à exprimer et à discuter. Le jeu doit alors se fonder d'une part sur une base de données cataloguant et rendant facilement accessibles et éditables les items du jeu (y compris les règles, conçues dès lors comme des éléments de contenu éditables) et d'autre part le forum.

Les joueurs aussi peuvent chercher et utiliser les items pour jouer, et les commenter - par exemple pour confronter leur expériences, échanger des astuces - et donc aussi accéder à (une partie de) ce forum. Ils peuvent discuter, entre eux et avec les concepteurs, sur les règles d'une action, le gain et la perte des points. Une marque permet de repérer si les messages sont postés par un joueur ou par un concepteur; les concepteurs peuvent s'ils le souhaitent masquer certains de leurs échanges aux joueurs.
Ces différentes exigences – espace virtuel participatif, prolifération et évolutivité des données – concrètement demandent que l'on dispose d'une base de donnée et d'une infrastructure technique à base de services qui permette effectivement de remplir ces conditions.

3. L'exemple de SOS-21

Le jeu SOS-21 actuel est un « jeu sérieux » sur le Web s'adressant au grand public favorisant l'émergence et l'assimilation de bonnes pratiques de développement durable Ce jeu facilite de multiples passages du virtuel au réel en proposant des scènes en correspondance avec des territoires réels. Actuellement, plus de 80 scènes de jeu sont disponibles[2] en accès gratuit via le Web, localisées à Paris, en Bretagne, en PACA et dans plusieurs pays méditerranéens Les contenus sont constamment alimentés par une approche pédagogique visant à passer de la sensibilisation à l'action concrète. Techniquement, SOS-21 est un univers virtuel (« metaverse ») basse consommation (3D isométrique), accessible gratuitement sur le Web depuis un simple visualiseur Flash. Cet univers s'enrichit continuellement et notre équipe travaille en partenariat sur certains aspects du jeu pour rendre le jeu davantage participatif.

Le joueur pilote son personnage dans un univers virtuel reflétant des lieux qu'il peut connaitre (de sa région ou de sa ville par exemple). Les objets du jeu sont des éléments du zonage, des scènes, des documents multimédias joints, et les règles décrivent par exemple comment le joueur monte en niveau, gagne ou perd des points dans les différentes dimensions du développement durable, suivant les actions, les lieux et les décisions prises par le joueur. Les règles du jeu sont destinées à mieux appréhender l'arrière plan complexe des bonnes pratiques en développement durable pour les membres d'un public large, en leur permettant de retrouver leurs actions de tous les jours. A travers les situations dont ils sont acteurs, les décisions des joueurs prennent en compte les critères liés aux objectifs personnels et à la complexité des dimensions économiques, sociales et environnementales du développement durable, mettant en évidence des « design patterns » tels que le lien entre « penser globalement et agir localement » (Seffah *et al.,* 2011). L'architecture de participation doit pouvoir servir à des« réseaux de débats » et rejoindre des expériences analogues de « web d'idées » qui ont commencé à se développer (Buckingham, 2008) (Park, 2008) comme Cohere[3] , visant aussi ce domaine du développement durable.

Plus précisément, dans SOS-21 les partenaires d'une association, d'une région ou d'une ville souhaitant sensibiliser via le jeu aux priorités de « l'Agenda 21 » local,

[2] Voir http://www.sos-21.com/Enter-the-game.html et http://www.defismed.com/embarquement.html

[3] http://cohere.open.ac.uk/

devraient pouvoir eux-mêmes en tant que co-concepteurs créer et modifier facilement des scènes placées sous leur responsabilité. Autrement dit, ils devraient pouvoir éditer des items de ces scènes (comme des lieux, des décors, des règles ou des micro-jeux transposant des situations propres à leur cas) par des formes de prototypage rapide accessibles aux non-informaticiens. A noter qu'il est possible d'utiliser le jeu SOS-21 dans un cadre d'apprentissage : les joueurs sont alors des apprenants suivis par des enseignants. Ces derniers peuvent éditer les items dans un but pédagogique (par exemple disposer des « quizz » dans l'espace 3D, décider des gains ou pertes consécutifs aux actions des joueurs, etc. Comme tout concepteur, les enseignants s'appuient sur les dispositifs de participation tels que le forum présenté ci-après (cf. figure 7) pour la discussion des règles.



Figure 2. Ramasser une bouteille sur la plage dans le jeu SOS-21 (copies d'écran)

Le domaine du développement durable se prête particulièrement bien à la co-conception. En effet, ce domaine, marqué par la complexité, requiert constamment de débattre y compris de ce qui pourrait passer pour des évidences. Par exemple, le choix de faire gagner des points environnementaux lors de l'achat d'un yaourt dans un pot en verre (recyclable), pourrait faire l'objet d'un débat : « Ne doit-on pas prendre en compte le coût CO2 de la production et du recyclage du produit ? », « Ne doit-on pas s'assurer que le personnage triera effectivement le pot en verre ? ».

Par ailleurs, pour transformer le jeu existant en jeu multi-joueurs, nous avons décidé dans un premier temps de fournir au joueur la conscience (*awareness*) :

- des mouvements et des actions des autres joueurs présents au même endroit et au même moment,
- des effets persistants sur le monde virtuel des actions de l'ensemble des joueurs.

4. Prototypage du serveur et maquettage des logiciels clients

4.1 Choix de l'infrastructure

Pour mettre en place des mécanismes de conscience mutuelle entre les joueurs, nous avons tout d'abord pensé réutiliser un serveur générique et open-source de jeu massivement multi-joueurs. Nous avons ainsi étudié *RedDwarf* (initialement développé par *Sun* sous le nom de « DarkStar »).

Le principal résultat de cette étude a été de nous faire découvrir que les plateformes de ce type sont en fait des cadres applicatifs (*frameworks*) dont les fonctionnalités sont essentiellement la malléabilité des modèles de données, le passage à l'échelle de l'écriture et de la lecture de données, la notification des clients par le serveur et enfin l'exécution régulière de tâches sur le serveur. La gestion des règles de jeu, quant à elle,

ne fait pas partie de ce noyau et les tentatives de développement de modules génériques de gestion de jeux multi-joueurs ne nous ont pas semblé très matures. Par ailleurs, le protocole utilisé pour interroger *RedDwarf* (protocole binaire *ad hoc*) ne facilitait pas l'intégration avec notre plateforme Web de co-conception.

Notre recherche d'infrastructure logicielle s'est donc recentrée sur un service de persistance de données indépendant du domaine du jeu, mais facilement intégrable et présentant les mêmes propriétés que *RedDwarf*. À l'exception de l'exécution programmée, *CouchDB* répondait à chacune de ces exigences :

- Parce que ses données sont « sans schéma »[4], on peut ajouter des attributs librement et donc prototyper aisément une nouvelle fonctionnalité (y compris sur le serveur de production).
- Parce que son API utilise le protocole HTTP, son intégration est possible à partir de la plupart des langages existants.
- Parce que son API respecte les principes[5] de « REST » (Fielding, 2000), elle profite des « bonnes pratiques » de conception qui ont permis le passage à l'échelle du protocole HTTP.
- Parce que ses « vues » sont définies selon le patron de conception *MapReduce,* elles profitent notamment de la mise en cache des résultats intermédiaires.
- Enfin, *CouchDB* propose une fonctionnalité de notification des modifications[6].

4.2 Conscience mutuelle asynchrone

Comme nous l'avons vu précédemment nous souhaitons faire prendre conscience aux joueurs qu'ils partagent le même monde, que l'état de ce monde est le résultat des actions de chacun. Cette fonctionnalité, cependant, pose une difficulté de conception technique : si cet état est une ressource partagée par tous les joueurs et modifiée lors de chaque action, ces modifications effectuées en parallèle risquent de rentrer en conflit les unes avec les autres. *CouchDB* est capable de détecter de tels conflits, cependant la résolution de conflits va causer un délai dans la prise en compte des modifications et donc augmenter le risque de rencontrer d'autres conflits.

---

[4] *CouchDB* est une base de données dite « NoSQL », c'est-à-dire proposant une alternative aux systèmes de gestion de bases de données relationnelles. Les performances des systèmes relationnels reposent sur la construction automatique d'index en fonction du schéma de la base et des données elles-mêmes. La qualité de ces index décroissant au fur et à mesure de la modification des données, il est nécessaire quand cette qualité devient trop basse de reconstruire l'index, tâche extrêmement coûteuse en temps et en ressources. Pour cette raison, alors que ces systèmes remplissaient parfaitement leur office depuis quarante ans pour le système d'information des entreprises, elles ont dû laisser la place à des alternatives radicales dans les applications massivement participatives dans lesquelles les modifications sont incessantes (Facebook, Twitter, Amazon, etc.).

[5] Comme son nom l'indique, le style d'architecture *REST* (« Representational State Transfer ») se caractérise par l'échange des états complets d'une ressource (ou tout au moins de sa « représentation »). Une première conséquence est que le serveur n'a pas à mémoriser l'état du client (session). Une seconde conséquence est que les opérations du protocole peuvent être génériques (par exemple : lire, écrire, supprimer), et qu'il est alors possible d'utiliser uniquement des opérations « sûres » (sans effet sur le serveur et dont le résultat peut ainsi être mis en cache) ou des opérations « idempotentes » (pour lesquelles la réémission en cas de temps d'attente trop long ne pourra pas avoir des effets imprévus).

[6] CouchDB émule la notification du client par le serveur, impossible en HTTP, par la méthode dite de la « longue scrutation » (*long polling*) : le client a l'initiative de demander au serveur s'il y a eu une modification, mais celui-ci attend qu'il y en ait une pour lui répondre.

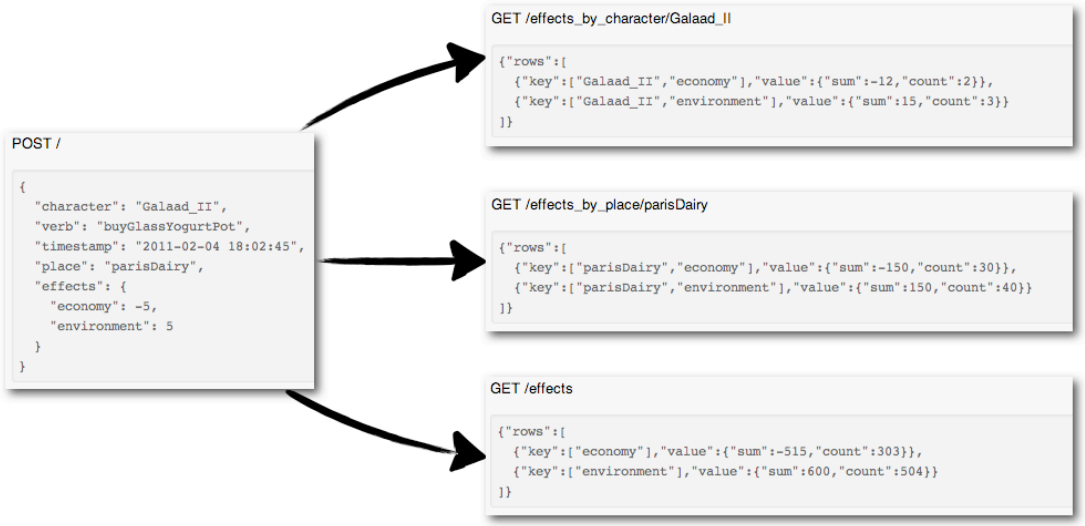


Figure 3. Impact de l'action d'un joueur sur son score, sur l'état de la scène et sur l'état du monde (exemples de requêtes envoyées au serveur)

Pour éviter de rentrer dans un tel cercle vicieux, il est conseillé de suivre la « recette du banquier » (Anderson *et al.*, 2010). De même que dans le système d'information d'une banque les véritables données sont les transactions alors que la balance des comptes n'est qu'une « vue » sur ces données, de même l'état du monde, des scènes et les scores des joueurs ne sont que des vues sur les actions des joueurs (cf. Figure 3). La seule différence entre les systèmes de type *MapReduce* et les systèmes traditionnels des banques est que ces « vues » ne sont pas calculées pendant la nuit mais « à la demande », en réutilisant les résultats intermédiaires non-impactés par les modifications survenues depuis la dernière consultation.

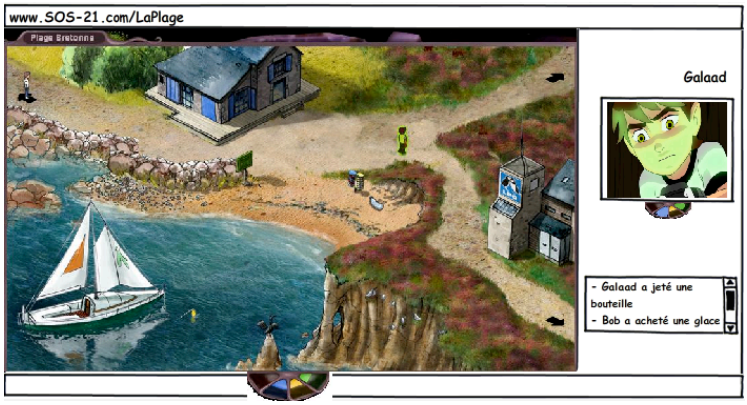


Figure 4. Score des personnages présents dans la scène

Les trois vues aujourd'hui prototypées dans le serveur pourront prendre des formes différentes dans le client :

- en plus de son propre score, le joueur pourra consulter le score des personnages présents dans la même scène, comme c'est le cas sur la Figure 4 : sur son écran

de jeu, le joueur dont le personnage est « Bob » connaît en permanence ses scores (en bas de l'écran), et peut demander à voir le score d'autres personnages (par exemple ceux de « Galaad » qui s'affichent alors dans le volet latéral)

- l'état du monde et des scènes pourra apparaître sous forme de jauges et éventuellement d'altérations de l'aspect de la scène (cf. Figure 5).



Figure 5. Score du personnage joueur (en bas) et score de la scène (à droite)

4.3 Conscience mutuelle synchrone

L'étape suivante consiste à donner conscience au joueur des actions et des mouvements ayant lieu au même moment dans la scène où il se trouve. CouchDB propose un mécanisme de notification des changements en fonction d'un filtre défini par le développeur. Dans notre cas, il s'agit donc d'un filtre sur le lieu (cf. Figure 6).

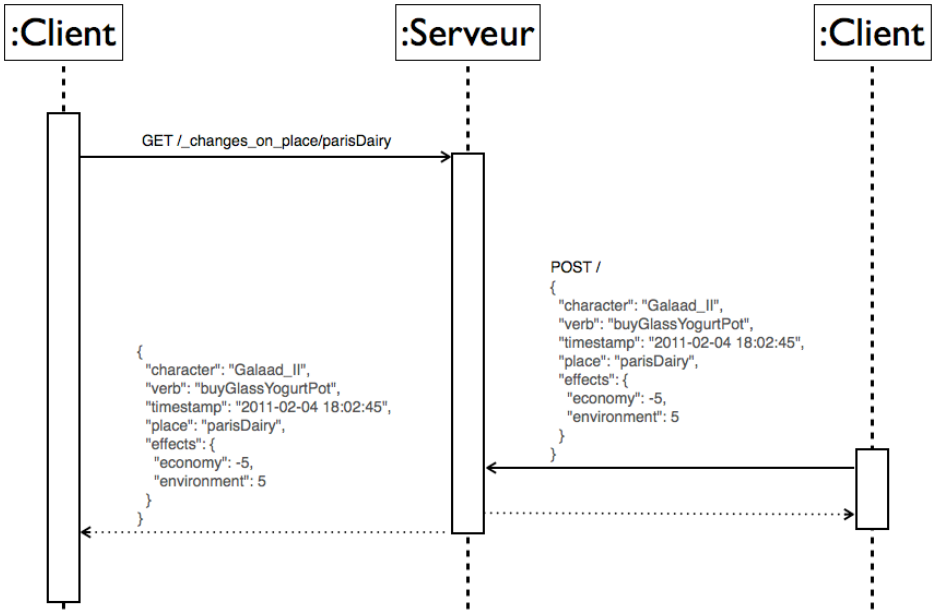


Figure 6. Notification des évènements ayant lieu dans une scène (diagramme de séquence UML)

Si l'on gère les déplacements en mettant à jour un objet correspondant au personnage[7], le service de notification utilisé pour les actions semble fonctionner également pour les déplacements. Cependant deux cas particuliers doivent être pris en compte :

- lorsque le joueur change de scène, la notification doit concerner non seulement la nouvelle scène mais aussi l'ancienne afin de faire disparaître le personnage,
- lorsque le joueur n'est plus connecté (car, même s'il reste « dans » la scène jusqu'à la prochaine connexion, il ne doit plus apparaître).

Le deuxième cas nous amène à considérer également le cas où le client se serait subitement arrêté sans se déconnecter. La solution habituelle à ce problème dans les jeux massivement multi-joueurs est de déconnecter automatiquement les personnages après quelques minutes d'inaction. Étant donné que *CouchDB* ne permet pas d'exécuter des tâches à intervalle régulier, nous avons mis en place, à destination d'un client spécifique, une vue donnant la liste des personnages connectés mais dans l'inaction depuis un moment donné.

4.4 Édition, consultation et discussion des règles

Pour que les règles puissent être conçues collectivement, celles-ci doivent être partagées et modifiables par les utilisateurs autorisés à le faire. La mise en place de la plateforme Web de co-conception participative (cf. Figure 7) nécessite donc d'une part de déporter les règles du client vers le serveur[8] et d'autre part de les considérer comme des données plutôt que comme des programmes.

Dès lors, la gestion des actions présentée plus haut doit être modifiée de sorte que le client n'envoie que l'identifiant de l'action et que ce soit le serveur qui ajoute à la transaction les bonus et malus correspondants (cf. Figure 8).

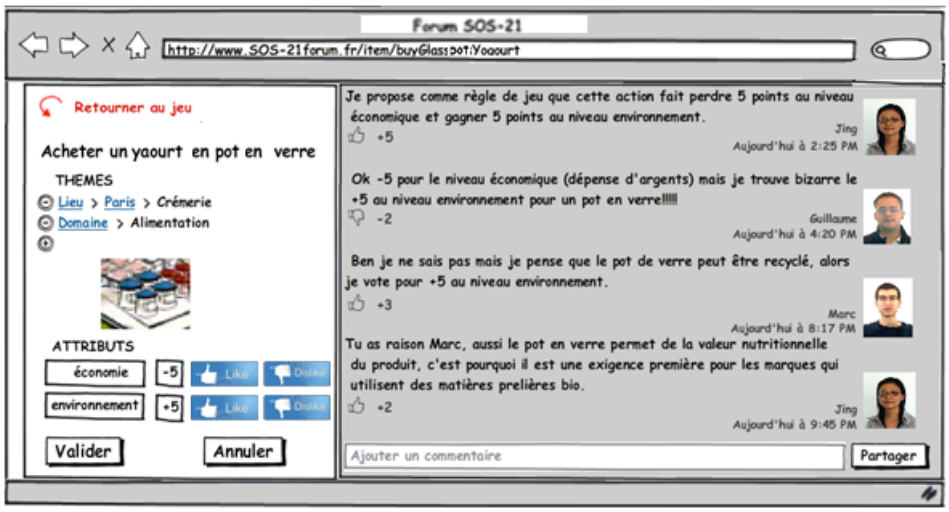


Figure 7. Discussion des règles du jeu (maquette)

[7] La « recette du banquier » est ici inutile puisque normalement chaque personnage n'est contrôlé que par un seul joueur.

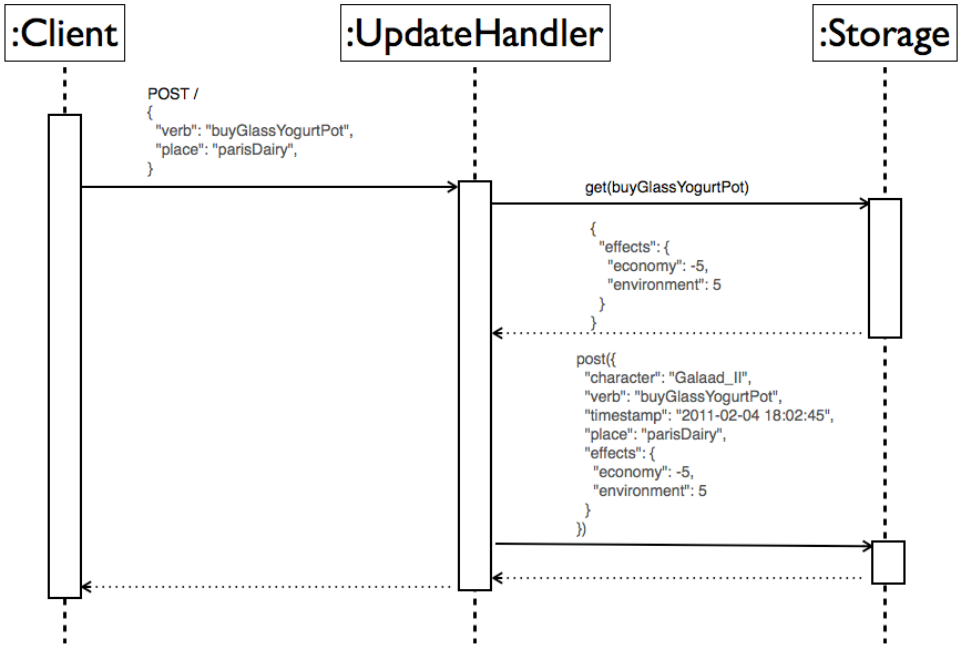


Figure 8. Gérer les règles comme des données (diagramme de séquence UML)

5. Perspectives et conclusion

Dans cet article, nous avons précisé la notion de jeux sérieux massivement participatifs et proposé des éléments de solution fonctionnels et techniques, reposant sur une base de données NoSQL (CouchDB) en indiquant sur quelques exemples pourquoi cette solution conviendrait mieux au service des jeux sérieux participatifs. Cette réflexion va de pair avec la mise en place en parallèle d'une architecture nous permettant de valider progressivement, dans le même mouvement certains aspects proposés (cf. §4). L'approche retenue permet de gérer les actions des joueurs et de les notifier des changements qui surviennent localement et globalement, grâce aux vues. Elle favorise la structuration des connaissances ajoutées dans le jeu, en particulier les objets des scènes et les règles, de façon à ce que ceux-ci soient plus faciles à retrouver et discuter dans une communauté large.

La suite de notre travail consiste notamment à mieux intégrer les aspects géographiques du jeu SOS-21, auquel s'ajoutent régulièrement des partenaires en France et en Europe et à mieux intégrer les règles du jeu dans CouchDB, en implantant la solution du Forum présentée en maquette sur la figure 7. Nous allons classer les joueurs selon le score pour savoir favoriser la compétition entre eux et les motiver en jouant. L'interface du jeu devrait aussi pouvoir présenter l'activité des participants en restituant leurs multiples points de vue, contribuant ainsi à augmenter le potentiel d'enquête des participants et « l'intelligence qualitative », comme nous l'avons montré dans de précédents travaux (Zacklad *et al.* 2008).

---

[8] Cela permet en outre d'éviter que les joueurs puissent tricher en envoyant au serveur des requêtes qui ne correspondraient pas aux règles du jeu.

Par ailleurs, nous souhaitons dès que possible proposer à des groupes de co-concepteurs non-informaticiens d'éditer le jeu, de créer et modifier au fil de l'eau leurs propres scènes et leurs propres règles, selon un mode de prototypage rapide et en discutant entre eux, au fur et à mesure, du contenu des règles du jeu. Il sera également intéressant de savoir ce que les –joueurs et concepteurs- pensent du jeu et des dispositifs participatifs proposés, et comment ils les utilisent effectivement. Pour cela, nous utiliserons une approche de « living lab » pour observer en temps réel ces participants et de perfectionner le jeu de façon dynamique en fonction des leçons de ces observations.

Remerciements

Cette publication bénéficie du concours du Ministère français de l'Enseignement Supérieur et de la Recherche, dans le cadre de la thèse de Nour El Mawas commencée en septembre 2010 à l'UTT.

Références

Anderson J. Chris, Lehnardt Jan, Slater Noah. « *CouchDB: The definitive guide* » O'Reilly editor, January 2010

Buckingham, S. "Cohere: Towards Web 2.0 Argumentation", Proceeding of the 2008 conference on Computational Models of Argument: COMMA 2008, Pages: 97-108, 2008, ISBN ~ ISSN:0922-6389 , 978-1-58603-859-5

Cahier J.-P. (2008)"Les leviers du mode communautaire", in Revue Archimag, Guide pratique n°32 " Collaboratif et gestion des connaissances ", janvier-février 2008, pp.16-24 (*en ligne* : HTTP://cahier.tech-cico.fr/int/jpctcao08.pdf)

Ehn, P. Participatory Design and the Collective Designer. (with Richard Badham) in Proceedings of Participatory Design 2002, Malmö, 23-25 June 2002

Fielding R., "Architectural Styles and the Design of Network-based Software Architectures", thesis at university of California, Irvine, 2000, Chapter 5

O'Reilly, T.: What is web 2.0: Design patterns and business models for the next generation of software. Social Science Research Network (2005)

Park, J., Topic Maps, Dashboards and Sensemaking, proceedings of Topic Maps Research and Applications conference (TMRA 2008)

Richardson L., Ruby, S.,. « *RESTful Web Services* », O'Reilly editor may 2007)

Seffah, A., Cahier J.-P., Benel, A. "Thinking Global Acting Local: A Pattern to Designing information-Intensive Services for Global Crisis Management", 8th International Conference on Information Systems for Crisis Response and Management ISCRAM," May 8-11 2011, Lisbon, Portugal

Winograd Terry, Flores Fernando. Understanding Computers and Cognition. Addison-Wesley, USA, 1986.

Zacklad M., Cahier J.P., Bénel A., Zaher L.H., Lejeune C., Zhou C., Intelligence qualitative dans les situations à risque complexes et apports du web socio-sémantique, WISG'08 (Workshop Interdisciplinaire sur la Sécurité Globale),Troyes 29-30/01/2008.

Zacklad, M. Communities of Action: a Cognitive and Social Approach to the Design of CSCW Systems In Proceedings of GROUP'2003, pp. 190-197, Sanibel Island, Florida, USA.

Zhou C., Bénel A., Lejeune C.."Towards a standard protocol for community-driven organizations of knowledge" Proceedings of the 13th int. conf. on Concurrent Engineering, Antibes, September 18-22, 2006. pp. 438-449. Frontiers in Artificial Intelligence and Appl. vol. 143. Amsterdam: IOS Press. 2006.