

Fabrique logicielle de réseaux sociaux spécialisés (aspects fonctionnels)

Benjamin Billet*, David Fernandez*, Didier Parigot*

*Équipe-projet ZENITH, Inria
{prenom.nom}@inria.fr

1 Introduction

En partenariat avec la startup BEEPEERS¹, nous concevons une fabrique logicielle (Greenfield et Short, 2003) pour le développement de réseaux sociaux spécialisés à destination de communautés ciblées. L'objectif de cette fabrique est de minimiser les coûts de conception et de production de ces réseaux. Concrètement, cette fabrique opère par spécialisation d'un réseau social abstrait, au moyen d'un mécanisme de sous-typage et de requêtes génériques pour obtenir les réseaux sociaux spécialisés.

2 fonctionnalités de la fabrique logicielle

Spécialisation Un **réseau social spécialisé** est un réseau à destination d'une communauté spécifique (p. ex. les exposants et visiteurs d'un salon, ou encore, les licenciés d'un club sportif). Le vocabulaire et les fonctionnalités d'un tel réseau sont conditionés par les besoins de cette communauté. À l'aide de l'expertise de BEEPEERS, nous avons réalisé un **réseau social abstrait**² regroupant tous les concepts utilisés par leurs différents réseaux sociaux. L'obtention des nouveaux réseaux sociaux spécialisés se fait en spécialisant ce réseau abstrait au moyen de fichiers de configuration. Compte tenu de la forte connectivité des données manipulées par ces réseaux, ils reposent sur des **bases de données graphe**.

Généricité Un mécanisme de **typage** et de **sous-typage** construit au dessus des bases de données graphes permet d'étendre les concepts du réseau social abstrait pour définir un réseau social spécialisé donné. Concrètement, un sous-type héritant d'un type de nœud du réseau abstrait pourra (i) renommer ce concept, et (ii) réduire la liste des fonctionnalités qui lui sont associées. Les types de nœud ainsi redéfinis par les réseaux sociaux spécialisés, forment une hiérarchie de types de laquelle sera déduits le schéma de la base de données graphe.

Il est important de disposer de requêtes exécutables sur tous les réseaux. Nous avons donc élaboré un moteur de requête qui supporte ce mécanisme de typage et sous-typage. Ainsi, les **requêtes génériques** définis à l'aide des types de nœud du réseau social abstrait, fonctionnent aussi avec leurs sous-types définis par les réseaux sociaux spécialisés.

1. www.beepeers.com (07/10/2015)

2. Schéma du modèle de données du réseau social abstrait : <https://huit.re/eiZDMvh7> (07/10/2015).

Système de Recommandation générique Nous avons établis, que pour mettre en œuvre les algorithmes usuels de la **recommandation** (Bobadilla et al., 2013) (p. ex. filtrage collaboratif) sous la forme de **data flow**, seul quatre opérateurs étaient nécessaires. Nous avons donc élaboré un système de data flow muni des opérateurs suivants : *PatternFinder*, *Product*, *Map* et *Aggregator*. Nous avons conçu un langage dédié permettant de décrire ces data flow afin d’instancier les algorithmes de recommandation. Comme ces algorithmes utilisent des requêtes génériques, ils sont exploitables par tous les réseaux sociaux spécialisés.

3 Conclusion

Nous venons de présenter les fonctionnalités requises par une fabrique logicielle pour la production de réseaux sociaux spécialisés : mécanismes de typage, moteur de requêtes et système de recommandation génériques. À l’heure actuelle, un prototype de notre fabrique implémente toutes ces fonctionnalités. Des tests de validation ont été effectués sur l’ensemble des données des applications de BEEPEERS. Les fonctionnalités ajoutés aux bases de données graphe (typage, sous-typage et requêtes génériques) ont été implémentées au dessus de Blueprints (un ensemble d’interfaces de manipulation de graphe supporté par les principales bases de données graphe). Le système de recommandation, qu’en à lui, a été conçu au moyen de SON(Lahcen et Parigot, 2012) un intergiciel pour le développement d’applications orientés service à base de composants ce qui permet une exécution distribuée des dataflow. Par manque de place, nous n’avons pas détaillé l’implémentation.

Remerciements Nous remercions Mickaël Jurret de BEEPEERS pour sa collaboration active.

Références

- Bobadilla, J., F. Ortega, A. Hernando, et A. Gutiérrez (2013). Recommender systems survey. *Knowledge-Based Systems* 46.
- Greenfield, J. et K. Short (2003). Software factories : Assembling applications with patterns, models, frameworks and tools. In *Companion of the 18th Annual ACM SIGPLAN Conference on Object-oriented Programming, Systems, Languages, and Applications*.
- Lahcen, A. A. et D. Parigot (2012). A lightweight middleware for developing P2P applications with component and service-based principles. In *15th IEEE International Conference on Computational Science and Engineering*.

Summary

This paper introduces a software factory for developing social networks. This factory takes an abstract social network and creates a concrete one, using mechanisms such as sub-typing and behavior overloading.