

Fabrique logicielle de réseaux sociaux spécialisés : Aspects fonctionnels

Benjamin Billet*, David Fernandez*, Didier Parigot*

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

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 pour obtenir les réseaux sociaux spécialisés.

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) dont le vocabulaire et les fonctionnalités sont conditionnés 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 d'un nouveau réseau social se fait en spécialisant ce réseau abstrait au moyen de fichiers de configuration fournis à la fabrique. D'autre part, compte tenu de la forte connectivité des données manipulées par ces réseaux, ils sont conçus au dessus des concepts des **bases de données graphe**.

Plus précisément, un mécanisme de **typage** et de **sous-typage** construit au dessus des bases de données graphes permet d'hériter des concepts du réseau social abstrait pour définir un réseau social spécialisé. Concrètement, un sous-type héritant d'un type de nœud du réseau abstrait peut (i) renommer ce concept, et (ii) réduire la liste des fonctionnalités qui lui sont associées. Les types de nœud ainsi redéfinis pour un réseau social spécialisé, forment une hiérarchie de types qui définit le schéma de la base de données graphe pour ce réseau. Pour éviter un travail de développement spécifique à chaque réseau, 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. Ainsi, les **requêtes génériques** définis avec les types du réseau social abstrait fonctionneront pour tous les réseaux.

Nous avons établi que pour mettre en œuvre les algorithmes classiques de la **recommandation** (Bobadilla et al., 2013), p. ex. filtrage collaboratif, sous la forme de **data flow**, seuls

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

2. Schéma du modèle de données du réseau social abstrait : <https://huit.re/eizDMvh7> (07/10/2015).

Fabrique logicielle de réseaux sociaux spécialisés

quatre opérateurs sont nécessaires : *PatternFinder*, *Product*, *Map* et *Aggregator*. Un système de data flow muni de ces quatre opérateurs et son langage dédié ont donc été élaborés. Comme ces algorithmes utilisent des requêtes génériques, ils sont exploitables par tous les réseaux sociaux spécialisés.

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 ces fonctionnalités au dessus de Blueprints (Apache TinkerPop, 2015) : un ensemble d'interfaces de manipulation de graphes supporté par les principales bases de données graphe. Le système de recommandation, quant à lui, a été conçu au moyen de SON (Lahcen et Parigot, 2012) un intergiciel pour le développement d'applications orientées service à base de composants. Cela permet une exécution en parallèle et en pipeline des dataflow. Des tests effectués sur l'ensemble des données des applications de BEEPEERS nous ont permis de valider cette approche préliminaire.

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

Références

- Apache TinkerPop (2015). Tinkerpop3 Webpage. <http://tinkerpop.incubator.apache.org>.
- 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 mechanism of sub-typing.