

Synchronization Problems in Parallel Activity of SmartScribo Blog Processors

Rustam Kadirov, Ivan Galov, Dmitry Korzun

Department of Computer Science,
Petrozavodsk State University (PetrSU)
Petrozavodsk, Russia
{kadirov, galov, dkorzun}@cs.karelia.ru

Abstract

The SmartScribo system is a Smart-M3 application for multi-blogging. Blog data and their semantic relations to users are stored in the blogosphere smart space, which is shared by the users. In multi-blogging, blog data are simultaneously received and published by many users in many blogs at many blog services. SmartScribo blog processors perform communications with blog services on behalf of the users. In this extended abstract, we state specific synchronization problems that can happen due to parallel activity of blog processors. Experiments with SmartScribo indicate that though the degradation is possible it is not extraordinary in regular blogger activity. The required synchronization can be achieved with a small set of additional knowledge the blog processors share in the blogosphere smart space.

Index Terms: Multi-blogging, SmartScribo, Smart-M3, Synchronization.

The SmartScribo system is a Smart-M3 application that provides access to the blogosphere by means of shared smart spaces for dynamic linkage of bloggers and blog content [1]. SmartScribo users interact with multiple blogs at many blog services using their devices as clients. For a given user, her/his instant blog data and context are stored in the personal smart space. The latter can be shared between SmartScribo agents, surrounding Smart-M3 applications and users, so forming the blogosphere smart space [2].

A SmartScribo blog processor (BP) performs communications between the blogosphere smart space and assigned blog services. It is implemented as a Smart-M3 knowledge processor running on a server as a daemon process. The set of available BPs defines a SmartScribo infrastructure. The latter also includes a Smart-M3 semantic information broker (SIB) [3], which maintains the blogosphere smart space (Fig. 1).

A SmartScribo client accesses in the smart space those blogs that the user is interested in. The client publishes notifications about pertinent blog messages into the space [2]; Responsible BPs track notifications and contact blog-services. Recent SmartScribo release includes BP for LiveJournal and RSS-based BP [4]. The latter supports the read-only mode for various blog services. A BP for Twitter is in progress.

The basic operations any BP implements are the following.

- Authorization on the assigned blog service on behalf of the user. The login/password data are provided by the client either via the smart space or directly.
- Operations with user accounts on the blog service side (add, update, remove).
- Transit of blog messages (posts and comments) between the smart space and blog services (mediator for write/read operations).

The basic design solutions applied to a common SmartScribo BP we studied in [4]. Our recent focus is on the synchronization problems when several BPs operate simultaneously,

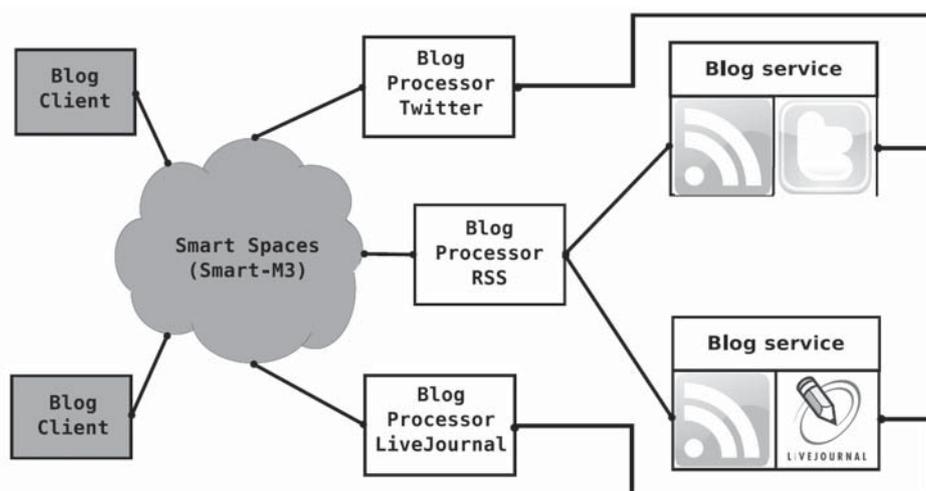


Fig. 1. Interaction of Blog Processors in SmartScribo

each serves a non-exclusive set of users. Note that the problems appear because of multi-blogging features.

- 1) *Blog service authorization.* Read/write access to a blog service requires authorization. If several BPs are available for the same service then there can be several independent attempts for authorization on behalf of the same person. Therefore, the information on which services the user is online must be synchronized between all responsible BPs, e.g., by its sharing in the space. Since the default rule is one BP per service (in read/write mode) the problem importance is not very high.
- 2) *Blog duplication in the space.* The same notification for blog data can be processed with several BPs. As a result, several BP can retrieve the same data from the service and publishes them in the space. Therefore, a mutual exclusion mechanism is needed with sharing appropriate knowledge in the space. In recent implementation, the problem leads to duplicated data stored in the space with different IDs. Consequently, clients also show the users the duplicates.
- 3) *Knowledge integrity.* SmartScribo allows dynamic scenarios when clients join and leave the blogosphere smart space as well as BPs become available or unavailable. The relation between users and their instant blog content must be preserved actual and consistent. A possible solution is an additional SmartScribo agent that regularly analyzes the space, remove redundant data and restore missed pieces. This solution improves the performance compared with a timeout mechanism on the client side, which the recent implementation provides.

REFERENCES

- [1] D. Zaiceva, I. Galov, and D. Korzun, "A blogging application for smart spaces," in *Proc. 9th Conf. of Open Innovations Framework Program FRUCT and 1st Regional MeeGo Summit Russia-Finland*, Apr. 2011, pp. 154-163.
- [2] I. Galov and D. Korzun, "Compositions of personal smart spaces in multi-blogging," in *Proc. 10th Conf. of Open Innovations Association FRUCT and 2nd Finnish-Russian Mobile Linux Summit*, Nov. 2011, pp. 54-58.
- [3] D. G. Korzun, S. I. Balandin, V. Luukkala, P. Liuha, and A. V. Gurtov, "Overview of Smart-M3 principles for application development," in *Proc. Congress on Information Systems and Technologies (IS&IT'11), Conf. Artificial Intelligence and Systems (AIS'11)*, vol. 4. Moscow: Physmathlit, Sep. 2011, pp. 64-71.
- [4] R. I. Kadirov and D. G. Korzun, "Smartscribo blog processor for multi-blogging in smart spaces," in *Proc. 10th Conf. of Open Innovations Association FRUCT and 2nd Finnish-Russian Mobile Linux Summit*, Nov. 2011, pp. 176-177.