

# PHEME: A Communication Infrastructure for Product Software Knowledge

Slinger Jansen, Sjaak Brinkkemper

Utrecht University

Information and Computing Sciences Institute

Utrecht, The Netherlands

{s.jansen, s.brinkkemper}@cs.uu.nl

## Abstract

Software vendors and system administrators experience similar problems when sharing software knowledge, such as software updates, commercial information, and software feedback, with end-users about software products. These problems incur high workloads for both software vendors and system administrators and often force them to use rigid and restricted methods of software knowledge delivery. PHEME<sup>1</sup>, a new infrastructure for software knowledge delivery, provides both software vendors and system administrators with a generic software knowledge delivery solution. By using the PHEME infrastructure, software vendors and system administrators can gain full control and flexibility of their software knowledge delivery process without much extra effort.

## 1 Software Knowledge Delivery

Product software knowledge is any type of knowledge that is shared between a vendor and end-user to keep a software product running and up-to-date. The vendor sends the end-user installation packages, software updates, software content, licenses, software manuals, and commercial information. The end-user sends usage feedback, error feedback, questions, configuration information, and requests for change to the vendor. The delivery of such knowledge is not a trivial task. End-users and vendors will often go through a complex software supply network, a system administrator, and a number of firewalls to reach the end-user. When trying to control the flow of product software knowledge both product software vendors and system administrators experience many problems.

Customer configuration updating (CCU) entails release, delivery, deployment, and usage and activation of product software [1]. Software vendors are proficient in building software products for their own problem domain, but do not wish to spend the same amount of effort on CCU tools. One

<sup>1</sup><http://www.cgbs.nl/PHEME/>

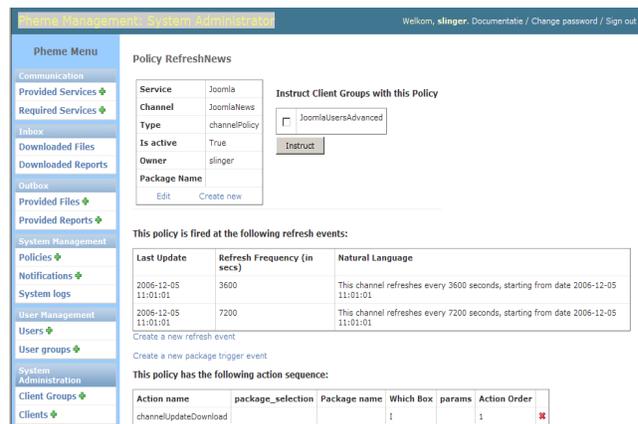


Figure 1. PHEME Screenshot

expects there to be many different tools available for the task of software knowledge delivery, especially since each and every product software vendor requires similar functions, such as delivering software knowledge, deploying a system in a potentially hostile environment, and sending and receiving feedback from end-users. Contrary to expectation, however, software vendors experience a complete lack of generic CCU tools [2].

System administrators, who act as a buffer between software vendors and users, are also finding many difficulties in acting as a hub between software vendor and end-user. They cannot remotely deploy software and they cannot control all information that enters and leaves the organization, with respect to software knowledge. Interesting to see is that both the software vendor and the system administrator wish to provide end-users with the most up-to-date software products in a secure and reliable manner and want the least possible overhead when it comes to knowledge delivery.

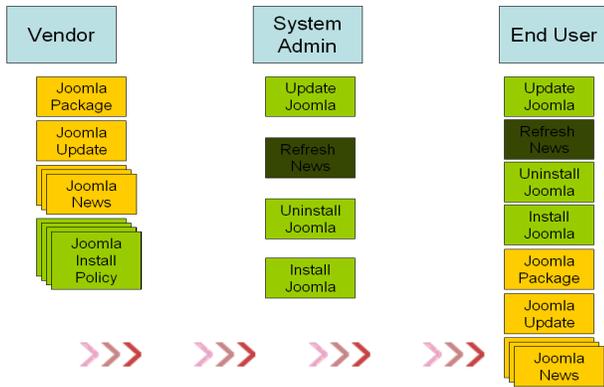


Figure 2. Three Participants

## 2 PHEME Infrastructure Overview

PHEME is an infrastructure that enables a software vendor to communicate about software products to end-users and enables system administrators to perform remote deployment, policy propagation, and policy adjustment. The infrastructure consists of a tool (PHEME), a protocol between software product and PHEME, a protocol between PHEMES, and a GUI. The PHEME tool is a server that resides on each system that acquires and distributes software knowledge through subscribe/unsubscribe channels. The server can accept and distribute all types of knowledge, including software knowledge delivery and deployment policies that describe behaviour of the PHEME tool. These policies can be manipulated securely and remotely.

PHEME enables software vendors to publish software knowledge in the form of licenses (for one end-user), software updates (for a group of end-users), and software news (for another group of end-users). PHEME enables customers to send knowledge in the form of usage, error, and support feedback. A system administrator can use PHEME to (with the right permissions) instruct other PHEMES, change and distribute delivery and deployment policies, control all communication between end-users and vendor, and redistribute software (knowledge). Finally, an end-user can edit policies, execute deployment policies (such as remove/install/update software product), determine when and how feedback will be sent to the vendor, and refresh all types of knowledge such as licenses.

## 3 Tool Demonstration

In this tool demonstration we show how PHEME can be used by a software vendor, a system administrator, and an end-user (see figure 2). The software vendor releases

software packages, product news, and policies of an open source package Joomla<sup>2</sup>. The system administrator instructs the end-user's system with deployment policies. The end-user applies the policies and deploys Joomla, demonstrating all types of customer vendor communication, such as product news.

PHEME can be used for software delivery in a software supply network. If product revisions are continuously released, using for instance a tool such as Sisyphus<sup>3</sup>, automatically uploaded to PHEME, and then automatically downloaded and deployed by the customer, the customer will always have the most up to date revision of the software product. Correct deployment can be ensured by the deployment policies in PHEME, or even by tools such as Nix<sup>4</sup>. PHEME is a fitting infrastructure for customer-vendor communication. To enable further integration of PHEME into a software vendor's organization, an interface is being developed between an open source CRM system and PHEME.

Even though no evaluation has been conducted, early experiments show that integrating with PHEME can be done quickly and is beneficial. PHEME provides the missing link in the release, delivery, deployment, and use and activation processes. Where Sisyphus provides a tool for releasing continuously, and Nix provides correct and complete deployment, PHEME can be used to deliver all types of software knowledge and also facilitate any usage and activation process. With PHEME in place, product software vendors can focus on software knowledge instead of knowledge delivery.

## References

- [1] S. Jansen and S. Brinkkemper. Definition and validation of the key process areas of release, delivery and deployment of product software vendors: turning the ugly duckling into a swan. In *proceedings of the International Conference on Software Maintenance (ICSM2006, Research track)*, September 2006.
- [2] S. Jansen, S. Brinkkemper, and G. Ballintijn. A process framework and typology for software product updaters. In *Ninth European Conference on Software Maintenance and Reengineering*, pages 265–274. IEEE, 2005.

## Acknowledgements

We would like to thank the Django and Joomla development teams for making such wonderful products that made this tool possible (in a shorter time than otherwise).

<sup>2</sup><http://www.joomla.org>

<sup>3</sup><http://sisyphus.sen.cwi.nl:8080/>

<sup>4</sup><http://www.cs.uu.nl/wiki/Trace/Nix>