HAL
open science

Weaving a semantic web across OSS repositories : a spotlight on bts-link, UDD, SWIM

Olivier Berger, Ion Valentin Vlasceanu, Christian Bac, Stéphane Lauriere

To cite this version:

Olivier Berger, Ion Valentin Vlasceanu, Christian Bac, Stéphane Lauriere. Weaving a semantic web across OSS repositories : a spotlight on bts-link, UDD, SWIM. WoPDaSD 2009 : 4th International Workshop on Public Data about Software Development, co-located with the OSS 2009 Conference, Jun 2009, Skövde, Sweden. hal-01369815

HAL Id: hal-01369815

https://hal.science/hal-01369815v1

Submitted on 21 Sep 2016

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.

Weaving a Semantic Web across OSS repositories: a spotlight on bts-link, UDD, SWIM

Olivier BERGER
Institut TELECOM, SudParis
9 rue Charles Fourier
91011 Évry Cedex, France
olivier.berger@it-sudparis.eu

Valentin VLASCEANU
Institut TELECOM, SudParis
9 rue Charles Fourier
91011 Évry Cedex, France
ion_valentin.vlasceanu@it-sudparis.eu

Christian BAC
Institut TELECOM, SudParis
9 rue Charles Fourier
91011 Évry Cedex, France
christian.bac@it-sudparis.eu

Stéphane LAURIERE
Mandriva S.A.
43, rue d'Aboukir
75002 PARIS, France
slauriere@mandriva.com

This work was conducted in the frame of the "System@tic Paris-Region" cluster (http://www.systematic-paris-region.org/), with funding of the Paris Region council.

ABSTRACT

Several public repositories and archives of facts about libre software projects, developed either by open source communities or by research communities, have been flourishing over the Web in the recent years. These enable new analysis and support new quality assurance tasks.

By using Semantic Web techniques, the databases containing data about open-source software projects development can be interconnected, hence letting OSS partakers identify resources, annotate them and further interlink them using dedicated properties, collectively designing a distributed semantic graph. Such links expressed with standard Semantic techniques are paving the way to new applications (including ones meant for "end-users"). For instance this may have an impact on the way research efforts are conducted (less fragmented), and could also be used by development communities to improve Quality Assurance tasks.

A goal of the research conducted within the HELIOS project, is to address bugtracker synchronization issues. For that, the potential of using Semantic Web technologies in navigating between many different bugtracker systems scattered all over the open source ecosystem is being investigated.

This position paper presents some existing tools, projects and models proposed by OSS actors that are complementary to research initiatives, and that are likely to lead to useful future developments: *UDD* (Ultimate Debian Database) and *bts-link*, developed by the Debian community, and *SWIM* (Semantic Web enabled Issue Manager) developed by Mandriva.

The HELIOS team welcomes comments on the future paths that can be considered in using the Semantic Web approach for improving these projects.

Keywords

RDF, forge, archive, bug, semantic, semantic web, ontology, database, repository of repositories, interoperability, bugtracker.

1. Introduction

The HELIOS project[1] is a joint project between academics and industrials in the frame of the Paris area System@tic cluster (under the "Libre software" thematics group), to build an *Applications Lifecycle Management* (ALM) open source platform.

Among other goals, HELIOS aims at addressing bugtracker synchronization issues[2]. To that purpose, the potential of using Semantic Web technologies for navigating between the many similar bugs filed in the different bugtracker systems scattered all over the open-source ecosystem has been experimented.

The first section introduces several open source tools illustrating the potential for semantically interconnected databases *UDD* (Ultimate Debian Database) and *bts-link*, developed by the Debian community, and *SWIM* (Semantic Web enabled Issue Manager) developed by Mandriva. The second section discusses new use-cases for researchers and open source practitioners, with the advent of more semantics in open source related software engineering facts repositories.

This will not constitute a detailed analysis nor the presentation of results achieved during the HELIOS project. The objective is mainly to attract attention to novel interesting projects, and present ideas that may trigger the interest of the research community, and maybe receive useful comments on the way the work done in the frame of HELIOS can be further shaped, and on how the use of such tools by the open source communities can be maximized.

Finally, a word of caution : this paper does not enter into the details of the Semantic Web approach, that some qualify as "the next revolution of the Internet"; it just focuses on the progressive adoption of S*emantic Web* concepts in various services and tools adopting interoperable representations of data through the use of standards such as RDF, RDFa, OWL, microformats and others. The reader unfamiliar with Semantic Web concepts and techniques is advised to read the gentle introduction presented at the previous edition of WOPDASD in [1].

---

[1] http://helios-platform.org/

[2] A more detailed description can be found at : https://picoforge.int-evry.fr/cgi-bin/twiki/view/Helios_wp3/Web/

2. Introducing several tools and services

We'll start by introducing the reader to some key projects that have been developed by the FLOSS communities recently, that will illustrate potential use of interconnected databases, with the help of Semantic Web techniques.

2.1 bts-link, a bug links watcher

Open-source GNU/Linux distributions such as Debian are composed of thousands of packages assembled (downstream) providing software which have been developed within hundreds of independent external projects (upstream).

Each GNU/Linux distribution maintains a central bugtracker (for instance debbugs running at `http://bugs.debian.org/`) open to reports from its users. In turn, each individual FLOSS project generally maintains a dedicated bugtracker (running Bugzilla, Mantis, Trac or others) mainly used by its developers, and sometimes hosted on a shared service like a software forge hosting many projects (like SourceForge). The bugs filed "downstream" by the end users of a distribution into its central bugtracker are most of the time related to the ones filed by the original developers "upstream", in their project's bugtracker.

It is the duty of the packagers (*package maintainers* in Debian terminology) to triage bugs, and to maintain the correspondence between "their" bugs in the distribution and the corresponding ones in the "upstream" bugtrackers of the projects.

In the Debian bugtracker `debbugs`, such a link is tracked by manually setting a "`forwarded-to`" attribute on a Debian bug (such links are publicly available).

`bts-link` [2] addresses the need for package *maintainers* in the Debian distribution to monitor *status* changes of the various "upstream" bugs that have been set as targets of `forwarded-to` links. Being alerted by `bts-link` of bugs being "closed" by the software's developers in an "upstream" bugtracker, the Debian maintainers can identify when it's the right time to prepare an updated package to be provided to Debian users in need of a fix.

The `bts-link` tool, running on a Debian distribution's server, periodically navigates such `forwarded-to` links, analyzes the bugs status by querying the bugtrackers interfaces, and eventually notifies the maintainers (and interested subscribers) whenever the linked bugs in "upstream" bugtrackers change state (open to closed, or closed to re-open, etc.)

The current way these bugs are interlinked with this "forwarded-to" attribute is somehow specific to the `debbugs` tool. Also, the interfaces and web services to access various bugtrackers contents are not standardized and sometimes some "screenscraping" is necessary on the Web pages.

In HELIOS, we would like to investigate the possibility of improving such a bts-link tool to make it less Debian-specific and to use Semantic Web techniques. We particularly hope to demonstrate benefits of LinkedData's best practices [3] to track links between such bugs in various bugtrackers, and the use of standard bug representations for interoperability. We hope that this will in the end prove beneficial for the Quality Assurance work in the whole open source ecosystem.

2.2 Ultimate Debian Database (UDD)

Some Debian Developers[3] have developed a repository called UDD, the "Ultimate Debian Database", for use inside the Debian distribution. This huge database, accessible to Debian contributors, groups *facts* about the Debian project, to ease the creation of (SQL) queries on what's happening in the Distribution. This is for instance very helpful for QA (Quality Assurance) tasks, like counting bugs with certain characteristics, or comparing packages in various ways.

In a sense, it is quite similar to the Flossmetrics[4] database or similar archives (aka RoR) well known to the academic community, which are collecting facts about many libre software projects, by extracting contents of the project data from the hosting forges.

We imagine UDD could be helpful to researchers through its integration with Flossmetrics and similar archives, as it contains facts about the packaging phases (downstream) of the libre software development process for the many software developed (upstream) in the forges (that have already been crawled in the archives). We imagine that analyzing links between such upstream and downstream activities of FLOSS actors can lead the way to new research.

But a general criticism that we could make on these databases is that their schema (the tables & columns layout, as well as the eventual relations) and the code of the data "harvesters" are the only means to understand the real semantics of the data collected there. There's not so much explicit semantics (unlike in RDF documents for instance), which diminish the possibility to cross-link facts between different databases. Sometimes the contents are even ambiguous between tables of the same database, for known reasons, because as explained by the UDD developers, there's actually much incoherence in some of the Debian tools already (although it still happens to deliver anyway).

As proposed in [1], using Semantic Web techniques would allow access to contents of such databases of facts using standard ontologies. That would help and convey some bits of commonly agreed semantics, hence fostering interoperability between these databases. This could be achieved in two ways: either by exposing the underlying data through a SPARQL access point on top of tools such as D2R [5], or by storing the data directly in an RDF database. The latter offers two advantages: first, its capability to harness all the relations available in the model by applying inference rules brings a deeper expressiveness to the data; second, it lets the data model evolve probably more easily than when using an RDBMS. However, the former relies on more tried and tested technologies, capable of handling queries against millions of raws across several tables with good performance[5], while the field of native triple databases is still in its infancy.

---

[3] Lucas Nussbaum, Stefano Zacchiroli and Marc Brockschmidt supervising the development made by Christian von Essen.

[4] http://melquiades.flossmetrics.org/

[5] We intend to demonstrate at the workshop a prototype of such a D2R server accessing the UDD database at least for access to bugs-related facts.

2.3 Semantic Web enabled Issue Manager (SWIM)

Semantic Web enabled Issue Manager (SWIM) is a recently rolled-out application[6] developed as a follow-up to the Nepomuk[7] project. It stems from the fact that, as stated by Henry Story[8]:

> "*Open source software is creating a global software space, with dependencies between projects, is meshing software from many different sources. But we are not meshing the data about the software!*"

SWIM aims at storing semantic statements pertaining to software engineering process. As of April 2009, it focuses on Mandriva bug descriptions by providing both automatically extracted data from a set of bug repositories (Mandriva bugzilla, KDE and Gnome bugzillas and others) and manual annotations enhancing the bug descriptions by knowledge that could not be inferred programmatically. SWIM semantic database can then be queried either from a dedicated KDE tool or from a dedicated Web interface. A SWIM KDE annotation tool features the capability of storing the annotations either locally (in case they relate to private tasks or in case of offline work) or publicly on the SWIM server.

Using the RDF standard and chosen ontologies, the semantic information is stored in a specialized data store, which maintains the relationship between the data. The structure of the data stored is described by an ontology based mostly on the data model[9] proposed by EvoOnt [4]. A similar approach is followed in [6], by applying in addition NLP technologies for extracting further information from the text data. In the near future, SWIM will be enhanced as well by text mining components, in particular in the context of the SCRIBO project [7]. SWIM will keep evolving within the open-source project MEPHISTO, which broadens the approach to software, hardware and people interconnections [8].

Providing an access to the UDD data with the use of a SWIM-interoperable service will lead to new analysis services harnessing the software engineering information system of open source software as a whole, comprised of the local information systems maintained by each project or each Linux distribution.

The HELIOS project is investigating the use of such techniques to try and manipulate data like bug reports for instance, and interconnect other tools such like `bts-link` with SWIM.

3. New use-cases and future work

The advent of the Web 3.0, i.e a Web of Linked Data [3] brings new perspectives to the field of software engineering in general, and to the tools developed within HELIOS in particular.

The LinkedData initiative is described as follow on the reference site:

> "*LinkedData is about using the Web to connect related data that wasn't previously linked, or using the Web to lower the barriers to linking data currently linked using other methods. More specifically, Wikipedia defines Linked Data as "*a term used to describe a recommended best practice for exposing, sharing, and connecting pieces of data, information, and knowledge on the Semantic Web using URIs and RDF*".*"

Like many other fields of application of the Semantic Web, the study and the management of open-source software engineering is likely to benefit tremendously from the interlinking of distributed structured data that the Linked Data paradigm is bringing. Its main realization could be the shift from closed *silos* of facts collected about open source projects (as in current databases and repositories of repositories) to future semantically-described Web resources inter-linked within an overlay graph of machine-processable data on top of existing open source software infrastructure (forges, bugtrackers, wikis, mailing-lists etc.).

3.1 Cross information systems bug tracking

The R&D efforts conducted in the frame of HELIOS on *bugtracker synchronization* aim at creating models and services that will allow to glue together various bug trackers by information pipelines realizing the vision of "porous federated containers" expressed by Mark Shuttleworth[10] [9]. The work will focus mostly on two parts:

- improving `bts-link` like tools so that the notification of linked bugs status changes can be used by more teams : not only only limited to Debian's debbugs on one side and some other few bugtrackers on the other side;
- starting from the model proposed by the SWIM project, a more advanced tool will be created, which will support use of data standardized in RDF formats such as EvoOnt's BOM, which will collect and manipulate more facts about bugs stored in different bugtrackers.

BOM (Bug Ontology Model) and similar ontologies like Baetle[11] will be complemented, so that they can better describe a generic bug model, as well as adapt to technical requirements of practical tools for open-source developers.

On an architectural level, once a generic data model is usable, the data conforming to it can be managed in two ways: either gather all the collected data into one giant (public or semi-public) RDF database (graph) providing a SPARQL endpoint for answering queries, or keep the data in the leaves of the network, connecting distributed semantic databases using the LinkedData model, hence embracing fully the Semantic Web vision.

We hope we can address the technical and computational challenges of such goals, and help foster interoperability between bugtracking services and client tools. We hope this can improve traceability of bugs and effort sharing between projects and distributions, hence improving the general quality of open source software.

3.2 Increased usability and data interchange

The use of Semantic Web techniques will become more widespread in the GNU/Linux systems in the coming years. Thanks to the outcome of the Nepomuk project, the recent Linux desktop include metadata management capabilities across applications, both in KDE (>= 4.2) and Gnome. Modern Linux

[6] http://club.mandriva.com/xwiki/bin/view/swim/

[7] http://nepomuk.semanticdesktop.org

[8] Semantic Web evangelist at Sun Microsystems

[9] http://www.ifi.uzh.ch/ddis/evo/

[10] the creator of the Ubuntu Linux project

[11] http://code.google.com/p/baetle/

desktops rely on standard components like Soprano or Tracker which are acting as building blocks of new end-user applications harnessing the potential of the Semantic Web approach, both at the personal and at he Web levels. Such applications may include for instance a BOM-aware bug triage client tool, a bug annotation tool, a distributed semantic search engine across software issues, etc.

The progressive adoption of semantic enabled databases on the Linux distributions servers [10] and within archives at research facilities, will increase the interoperability of all these services with the tools used by the end-users, developers or researchers on their desktops.

One of the challenges is the way standards can emerge from the various designs by researchers, developers and users, of ontologies modelling the realm of open source software development.

Another challenge relies on the capacity of handling efficiently very large volumes of data. Relational databases prove that the currently available data can be processed and can lead to useful knowledge and metrics assisting effectively the engineers in their production tasks, whereas the scalability of the Semantic Web approach in the field of software engineering remains to be experimented.

4. Conclusion

Efforts conducted in the frame of the HELIOS project aim at widening the use of developer-friendly applications harnessing the potential of Semantic Web aware services. Such applications include an improved bts-link, and other tools linking information across dedicated information systems, contributing to generic bridges federating OSS information systems into a giant knowledge base distributed over the Web.

The use of Semantic Web techniques is likely to improve the processes at stake in open-source software engineering and maintenance, both at the individual level through a deep integration of production and monitoring tools with the contributor's desktop and own mental vision of the processes, and at the collective level through better connections between distributed engineering facts in the open-source ecosystem, which are changing at rapid pace on the Web.

Not only will the Semantic Web techniques adoption ease the distributed engineering processes, but also the reuse of common data by both researchers and software developers or users, hence changing the way research on open-source software is conducted.

5. References

[1] Howison, J. 2008. *Cross-repository data linking with RDF and OWL - Towards common ontologies for representing FLOSS data.* Proceedings of the WOPDASD 2008 http://libresoft.es/oldsite/Activities/Research_activities/WoPDaSD2008_files/Paper3.pdf

[2] Berger, O. 2009. *Introduction to bts-link*. http://www-public.it-sudparis.eu/~berger_o/weblog/2009/02/05/introduction-to-bts-link-slides/

[3] Berners-Lee T. *Design Issues: Linked Data*. http://www.w3.org/DesignIssues/LinkedData.html

[4] Kiefer, C., Bernstein, A., Tappolet, J. *Mining Software Repositories with iSPARQL and a Software Evolution Ontology*. Proceedings of the ICSE International Workshop on Mining Software Repositories (MSR). Minneapolis, MA, May 19-20, 2007.

[5] Bizer, C.,Cyganiak, R. *D2R server - publishing relational databases on the Semantic Web* (poster). In Proceedings of the International Semantic Web Conference (ISWC). 2003.

[6] Damljanovic, D., Bontcheva, K. *Enhanced Semantic Access to Software Artefacts*. 4th International Workshop on Semantic Web Enabled Software Engineering (SWESE'08), in collaboration with ISWC 2008, Karlsruhe, Germany, October, 2008

[7] SCRIBO project – *Semi-automatic and Collaborative Retrieval of Information Based on Ontologies* – http://www.scribo.ws/

[8] MEPHISTO project. *Meshing People, Hardware and Software Together*– http://code.google.com/p/mephisto/

[9] *Shuttleworth urges Linux patch and bug collaboration* http://www.linux-watch.com/news/NS8470376604.html

[10] Tappolet J. June 2008. *Semantics-aware Software Project Repositories* ESWC 2008 Ph.D. Symposium