
	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

Document de spécification :  
Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie

<b>Livrable du au titre du projet</b>	HELIOS
<b>Lot</b>	WP3
<b>Tache</b>	3.1
<b>Livrable</b>	LI-15-RAP

<b>Rédacteur(s)</b>	<b>Vérificateur(s)</b>	<b>Approbateur(s)</b>
O. Berger, V. Vlasceanu	C. Bac	

<b>Documents de références</b>	
	Annexe technique du projet HELIOS

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

## Gestion des versions

N° de version	Date	Auteurs	Modification apportées
0.1	28/08/09	O. Berger	Initialisation du document
0.2	08/09/09	O. Berger	Mise à jour de la description des versions de composants
0.3	15/09/09	O. Berger	Compléments partie 2 (fonctionnalités) MAJ partie 1 (diagrammes ontologie)
1.0	16/09/09	O. Berger	Validation version majeure


**Copyright : ©2009 Institut TELECOM, O. BERGER, V. VLASCEANU**

**Ce document est diffusé sous les termes de la licence XXX à définir XXX, à l'exception de l'annexe « Helios\_BT reference documentation » qui est diffusée sous licence « Creative Commons Attribution 2.0 France ».**

**Ce document est le produit de travaux effectués dans le cadre du projet Helios, avec l'aide financière de la Région Ile-de-France.**

## Table des matières

1 Objectifs de ce document.....	4
1.1 Contexte : tâche 3.1 du projet Helios (Étude et modélisation).....	5
1.2 Structure du document.....	5
2 Bug modeling requirements.....	8
3 Review of some bug tracking systems.....	9
3.1 Bug attributes characterization.....	9
3.2 Bugzilla.....	10
3.3 Mantis.....	14
3.4 Debian's Debbugs.....	20
3.5 Launchpad's bugtracker.....	23
3.6 Gforge / FusionForge's tracker.....	26
3.7 Trac's bugtracker.....	28
4 Bug models.....	32
4.1 Baetle project.....	32
4.1 EvoOnt BOM.....	32
4.2 Our proposition: Helios_BT to complement BOM.....	37
4.3 Related ontologies and standards.....	41
5 Standardization.....	42

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

5.1 Demonstrators.....	42
5.2 Dissemination efforts.....	42
5.3 Future efforts.....	43
6 References.....	44
7 Définition du besoin.....	46
7.1 Rappels des éléments définis dans l'annexe technique.....	46
7.2 Compléments à l'issue d'une première phase d'étude.....	48
7.3 Contextes d'utilisation.....	48
8 Interfaces of a global bug monitoring system.....	54
8.1 Interfaces of the core generic system.....	54
8.2 First complementary applications.....	55
9 General architecture.....	57
10 Interfaces with bugtrackers.....	60
11 Integration with WP 2's documentation portal.....	61
12 Integration with the rest of the Helios platform.....	62
Annex : Helios_BT reference documentation.....	63
ANNEX : « Weaving a Semantic Web across OSS repositories: a spotlight on bts-link, UDD, SWIM ».....	100
12.2 bts-link, a bug links watcher.....	102
12.3 Ultimate Debian Database (UDD).....	103
12.4 Semantic Web enabled Issue Manager (SWIM).....	104
12.5 Cross information systems bug tracking .....	105
12.6 Increased usability and data interchange.....	105
ANNEX : « Bugtrackers triplification ».....	108
12.7 Introduction.....	108
12.8 Bug tracking on the Semantic Web.....	108
12.9 Current status.....	109
12.10 Conclusion.....	110
12.11 References.....	110



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie

Référence : LI-15-RAP

Version 1.0 du 16/09/09

## 1 Objectifs de ce document

Ce document présente le résultat de la tâche 3.1 « Étude et modélisation » du sous-projet 3 (WP3) portant sur la « Traçabilité des anomalies entre plateformes de développement ».

Le présent document (LI-15-RAP) contient donc deux grandes parties :

1. une **spécification d'un modèle générique de suivi d'anomalie** :

**Cette première partie présente les fonctionnalités des bugtrackers sélectionnés dans le cadre de l'étude, et propose un modèle générique de représentation de faits relatifs à un rapport de bug reprenant les caractéristiques communes à ces bugtrackers.**

**Une implémentation de ce modèle est proposée sous forme d'ontologie (appelée Helios\_BT, et basée sur EvoOnt BOM), décrite dans le langage OWL, afin de l'inscrire dans une perspective d'utilisation avec le langage RDF, comme standard du Web Sémantique.**

**Cela permet de proposer un format d'échange permettant de construire des applications interopérant avec les bugtrackers identifiés, notamment pour la réalisation d'un système de suivi générique d'anomalies (ci-dessous en 2ème partie).**


**Ceci offre des perspectives de standardisation des échanges entre bugtrackers, et donc d'avoir un impact large sur les outils utilisés notamment dans les communautés de développement Open Source. Une présentation des efforts de standardisation envisagés sera aussi fournie.**

2. une **description des interfaces sous formes de Web Services du suivi d'anomalie**

**Cette deuxième partie décrit un système générique de suivi de faits relatifs à des bugs saisis dans des bugtrackers externes, qui sera développé dans le cadre du projet Helios. Ce système vise à agréger dans une base de données unique et dans un format cohérent, dans une sémantique bien définie (ci-dessus en première partie), un ensemble d'informations relatives au suivi et au traitement de bugs qui permette de fournir :**

- **des interfaces d'accès et de suivi des bugs selon différents critères propres aux utilisateurs (interfaces Web ou APIs)**
- **un support pour l'expertise collaborative autour des bugs facilitant un certain nombre de tâches relatives à la maintenance des paquets logiciels (notamment via des fonctions d'annotation sémantique)**
- **un support pour la réalisation de premières applications utilisatrices, telles une version plus générique de l'outil bts-link, et des modules de suivi des bugs entre dépendances logicielles et au long d'un arbre de versions, intégrés dans la plateforme Helios (WP4)**

**Une description de l'architecture générale et d'un certain nombre de contraintes de réalisation sont également fournies.**

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

## 1.1 Contexte : tâche 3.1 du projet Helios (Étude et modélisation)

La tâche 3.1 du projet Helios a pour objet une étude qui «[...] proposera non seulement un modèle générique d'anomalie, mais aussi des interfaces sous forme de Web Services pour automatiser l'enchaînement des traitements associés aux anomalies. La description des inter-relations entre briques logicielles telle qu'on peut la trouver dans les gestionnaires de paquet logiciel utilisés dans les distributions libres (rpm, et deb) sera utilisée pour faciliter la recherche des relations entre les différents niveaux auxquels un bug peut s'exprimer. Cette étude proposera des scénarios pour analyser la circulation de l'information relative aux traitements des erreurs dans le cas de circulation ascendante (d'une brique utilisatrice vers une brique de base), descendante (d'une brique de base vers une brique utilisatrice), ou horizontale (entre briques de même niveau qui utilisent les mêmes briques de base). Ce modèle peut s'appuyer sur des études existantes comme Beatle ou les formats utilisés par des logiciels tels que ScmBug.

*L'un des livrables de cette tâche correspond à une proposition de norme de description d'un bug, ainsi que des interfaces d'interaction avec un gestionnaire de bug. Certains résultats du projet européen Edos pourront être réutilisés.»*


### Livrable(s) attendus :

- **LI-15-RAP** (dû à T0+8) *Spécification de modèle générique de suivi d'anomalie et une description des interfaces sous formes de WebServices du suivi d'anomalie*

## 1.2 Structure du document

Ce document est subdivisé en plusieurs chapitres :

- en partie 1 :
  1. Exposé des besoins en modelisation des bugs
  2. Présentation de différents gestionnaires de bugs
  3. Modèles de bugs
  4. Efforts de standardisation
- en partie 2 :
  1. Définition des besoins en matière de suivi d'anomalies et des contextes d'application étudiés
  2. Définition des interfaces et des fonctions détaillées attendues d'un système de suivi global des bugs
  3. Une description de l'architecture générale du système
  4. Une ébauche de liste d'interfaces avec des bugtrackers existants
  5. Une description des éléments relatifs à l'intégration avec les autres développements du projet HELIOS
- en annexes :

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

- documentation de référence de l'ontologieHelios\_BT
- article « Weaving a Semantic Web across OSS repositories: a spotlight on bts-link, UDD, SWIM » présenté au colloque WOPDASD 2009 en juin 2009
- article court « Bugtrackers triplification » soumis au challenge triplify 2009

Ce document « maître » est issu de l'assemblage de deux sous-documents qui ont été élaborés de façon indépendante afin de permettre une meilleure gestion des évolutions au cours de la rédaction :

1. LI-15-RAP-1 « Document de spécification de modèle générique de suivi d'anomalie »,
2. LI-15-RAP-2 « Document de spécification des interfaces Web Service du suivi d'anomalie »

Alors que le présent document LI15RAP est stabilisé pour assurer un état initial des spécifications du WP3 permettant d'initier la suite des tâches nécessaires au déroulement du sous-projet, chacun de ces deux sous-documents est amené à subir des révisions ultérieures, au cours du WP3, jusqu'à produire des versions définitives, soit respectivement :

1. Livrable LI-18-RAP : « Spécification de modèle générique d'anomalie »
2. Livrable LI-19-RAP : « Documentation des interfaces d'un service de suivi des anomalies »


Historique des versions des sous-documents intégrées :

Version LI15RAP stabilisé	Version LI15RAP1 intégrée	Version LI15RAP2 intégrée
1.0	0.7	0.2
0.3	0.7	0.2
0.2	0.6	0.1
0.1	0.5	0.1

### **Avertissement :**


**Certaines sections de ce document technique sont destinées à une audience de spécialistes supposés aptes à la lecture de documentation en anglais.**

**Elles sont donc rédigées en anglais, afin de faciliter sa diffusion directe auprès des communautés internationales concernées (développeurs d'outils de gestion de bugs, communautés Open-Source,...).**

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

## **PREMIERE PARTIE :**

# **Spécification d'un modèle générique de suivi d'anomalie.**

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

## 2 Bug modeling requirements

The context of this study is two-fold :

- specify the needs of bugs modeling in the frame of the Helios project in order to integrate a bugtracker (primarily Mantis, as selected in WP1) in the Helios platform
- propose a standard for bugs modeling in the frame of a more general use to foster interoperability between bugtrackers and other bugs management applications used in software development projects.

The primary focus here is *not* to study factors leading to software malfunctions or help to implement applications that would identify bugs in software code, or other software analysis in order to discover new bugs.

Instead, we seek to address the needs of software engineers and software production process in management of already identified bugs.

We adopt a pragmatic approach in trying to model existing properties and facilitate the improvement of existing applications, instead of reinventing completely novel models, in order to facilitate the deployment of applications that would rely on our model.

We primarily seek to support FLOSS software (due to the context of the Helios project), and propose a standard under terms of use that fit the needs of the Open Source communities.





Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence : LI-15-RAP
Version 1.0 du 16/09/09

### 3 Review of some bug tracking systems

This chapter is dedicated to a summary description of some popular Open Source bugtracking systems. The focus will be on bug properties that need to be considered in designing a bug model.

The studied bugtrackers are :

1. Bugzilla ( 3.2 )
2. Mantis ( 3.3 )<sup>1</sup>
3. Debbugs, the Debian project's bugtracker ( 3.4 )
4. Launchpad's tracker ( 3.5 )
5. GForge / Fusionforge tracker ( 3.6 )
6. Trac ( 3.7 )

Of course, there are many more bugtrackers, but the common properties of the first 6 tools studied are an excellent starting ground for a first bug model, which may be improved later when new bugtrackers are reviewed (in later tasks of the Helios project or in the future).

The format of each chapter will include :

- a short description of the bug tracking system (BTS),
- a description of the bug attributes in the tool (see below, in 3.1 ),
- a bug's lifecycle diagram.

Note that most of the descriptions provided in the following sections were borrowed from the respective bugtracker applications websites, and will not consist in an in-depth review of benefits or concerns with each tool's use.

#### 3.1 Bug attributes characterization


With each bugtracker description, we provide a standard cartridge which will summarize those attributes that have been identified in this study as relating to :

- **Bug description** : general bug properties (severity, description, date of submission, etc.)
- **Links bug / software** : to which versions of a software package a bug is related
- **Links bug / persons** : to which people/users a bug is related
- **Links bug / bugs** : to which other bugs a particular bug is linked
- **Miscellaneous** : other custom attributes

These categories will help map to the main entities modeled in the ontology presented in chap. 4 (Bug models).

---

<sup>1</sup> Among these, Mantis is the actual tool that has been selected to implement the Helios platform and is described in deliverable LI-02-RAP.

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

## 3.2 Bugzilla

The first BTS analyzed in this chapter is probably the most popular Open Source bugtracker **Bugzilla**, a Web-based general-purpose bugtracker tool originally developed and used by the Mozilla project.

Released as open source software by Netscape Communications in 1998, Bugzilla has been adopted by a variety of organizations for use as a defect tracker for both free software and proprietary products.

### 3.2.1 Short description of the BTS

According to the project website at <http://www.bugzilla.org/> ...

... Bugzilla is a Web-based general-purpose bugtracker tool originally developed and used by the Mozilla project. Released as open source software by Netscape Communications in 1998, Bugzilla has been adopted by a variety of organizations for use as a defect tracker for both free software and proprietary products.

Bugzilla is licensed under the Mozilla Public License. It was originally written by Terry Weissman in 1998 for the nascent Mozilla.org project, as an open source application to replace the in-house system then in use at Netscape Communications for tracking defects in the Netscape Communicator suite. Originally written in Tcl, Terry decided to port Bugzilla to Perl before its release as part of Netscape's early open source code drops, with the hopes that more people would be able to contribute to it as Perl seemed to be a more popular language at the time.

Bugzilla 2.0 was the result of that port to Perl, and the first version released to the public via anonymous CVS.

Bugzilla 3.0 was released on May 10<sup>th</sup> 2007 and brought refreshed UI, XML-RPC interface, custom fields and resolutions, mod\_perl support, shared saved searches, improved UTF8 support and others.

While the potential exists in the code to turn Bugzilla into a technical support ticket system, task management tool, or project management tool, Bugzilla's developers have chosen to focus on the task of designing a system to track software defects.

According to Bugzilla's site, the mandated design requirements include:

- the ability to run on freely available, open source tools. While Bugzilla development includes work to support commercial databases, tools, and operating systems, this is not intended to come at the expense of open source ones.
- the maintenance of speed and efficiency at all costs. One of Bugzilla's major attractions to developers is its lightweight implementation and speed, so calls into the



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence : LI-15-RAP
Version 1.0 du 16/09/09

database are minimized whenever possible, data fetching is kept as light as possible, and generation of heavy HTML is avoided.

- using ANSI SQL calls and data types in all new queries and tables. Avoid database specific calls and datatypes whenever possible. Existing SQL calls and data types should be converted to ANSI SQL.

While all the former may not be truly realized or matter for all development projects, Bugzilla remains one of the most popular bugtrackers used nowadays.

It is used in particular (most of the times with customized versions) by big projects such as commercial or open source GNU/Linux distributions (like Mandriva, RedHat) or development projects like KDE or Gnome all managing tens (or hundreds) of bugs.

### 3.2.2 Anatomy of a Bug

<b>Link bug-software:</b>	product and component
	platform and OS
	version
	target
<b>Link bug-persons:</b>	assigned to
	QA contact
	reporter
	CC list
<b>Link bug-bug</b>	URL
	dependencies
<b>Bug description</b>	summary
	status and resolution
	priority
	severity
<b>Miscellaneous</b>	status whitebord
	time tracking
	keywords
	attachments
	votes
	additional comments

Table 1: Bugzilla bug attributes

We reproduce here an excerpt of the bugzilla documentation that is published under the GNU FDL [6] :



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence : LI-15-RAP
Version 1.0 du 16/09/09

The core of Bugzilla is the screen which displays a particular bug. [...]. Fields marked \* may not be present on every installation of Bugzilla.

**1. Product and Component:** Bugs are divided up by Product and Component, with a Product having one or more Components in it.

[...]

**2. Status and Resolution:** These define exactly what state the bug is in - from not even being confirmed as a bug, through to being fixed and the fix confirmed by Quality Assurance.

**3. Assigned To:** The person responsible for fixing the bug.

**4. \*QA Contact:** The person responsible for quality assurance on this bug.

**5. \*URL:** A URL associated with the bug, if any.

**6. Summary:** A one-sentence summary of the problem.

**7. \*Status Whiteboard:** (a.k.a. Whiteboard) A free-form text area for adding short notes and tags to a bug.

**8. \*Keywords:** The administrator can define keywords which you can use to tag and categorise bugs - e.g. The Mozilla Project has keywords like crash and regression.

**9. Platform and OS:** These indicate the computing environment where the bug was found.

**10. Version:** The "Version" field is usually used for versions of a product which have been released, and is set to indicate which versions of a Component have the particular problem the bug report is about.

**11. Priority:** The bug assignee uses this field to prioritize his or her bugs. It's a good idea not to change this on other people's bugs.

**12. Severity:** This indicates how severe the problem is - from blocker ("application unusable") to trivial ("minor cosmetic issue"). You can also use this field to indicate whether a bug is an enhancement request.

**13. \*Target:** (a.k.a. Target Milestone) A future version by which the bug is to be fixed.

**14. Reporter:** The person who filed the bug.

**15. CC list:** A list of people who get mail when the bug changes.

**16. \*Time Tracking:** This form can be used for time tracking. To use this feature, you have to be blessed group membership specified by the "timetrackinggroup" parameter.

[...]

**17. Attachments:** You can attach files (e.g. testcases or patches) to bugs. If there are any attachments, they are listed in this section. Attachments are normally stored in the Bugzilla database, unless they are marked as Big Files, which are stored directly on disk.

**18. \*Dependencies:** If this bug cannot be fixed unless other bugs are fixed (depends on), or this bug stops other bugs being fixed (blocks), their numbers are recorded here.

**19. \*Votes:** Whether this bug has any votes.

**20. Additional Comments:** You can add your two cents to the bug discussion here, if you have something worthwhile to say.

The life-cycle, also known as work flow, of a bug is currently hardcoded into Bugzilla. **Figure 1** contains a graphical representation of this life cycle<sup>3</sup>. The workflow can also be expected to vary depending on the complexity of the bug, with a bug perhaps moving between a number of developers for different stages of resolution, followed by updates to the documentation by a technical writer before a tester can verify its resolution.

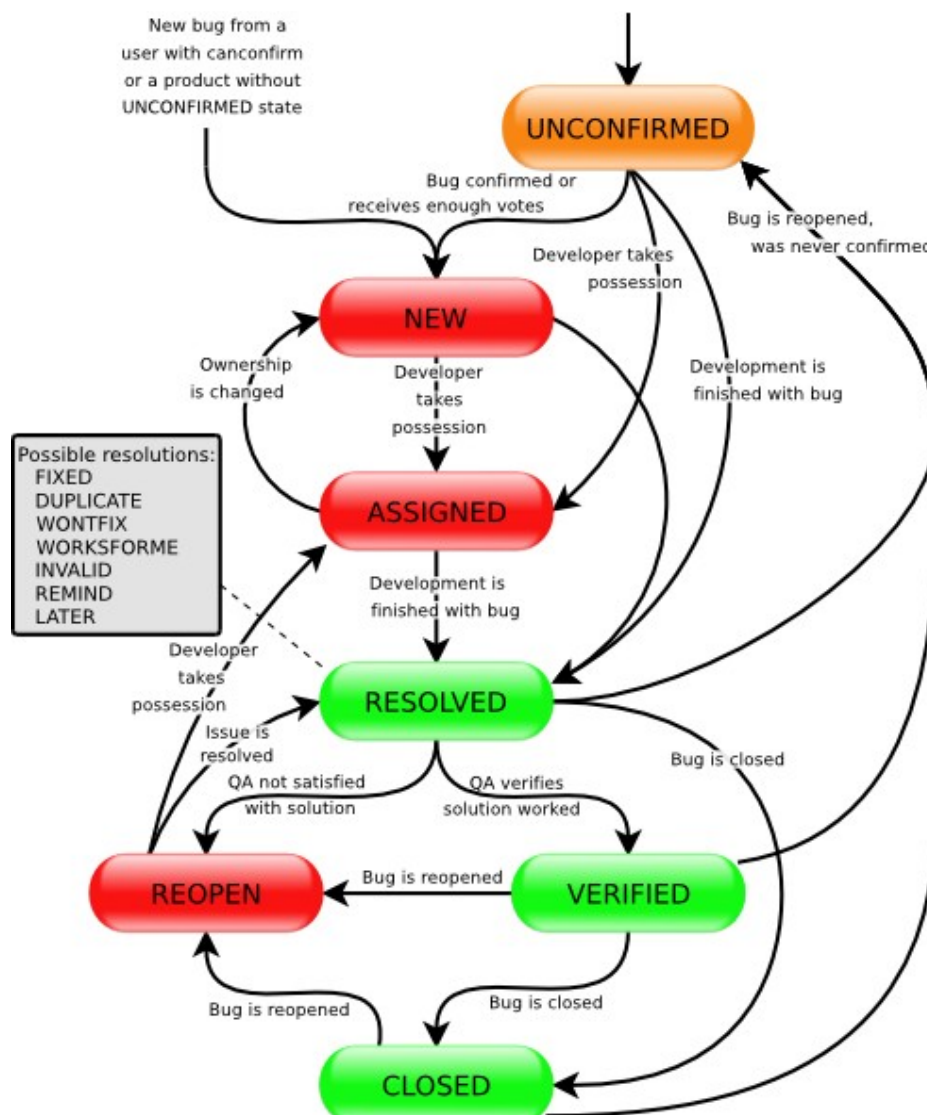


Figure 1: Lifecycle of a bug in Bugzilla

The following status represent the stages that a bug will move through in its lifetime:

- **New** - when a bug is newly created, it has a state “new”.

<sup>3</sup> The source of the diagram is the bugzilla documentaion [6]



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence : LI-15-RAP
Version 1.0 du 16/09/09

- **Assigned** - once the decision has been made to resolve a bug, it needs to be assigned to the person who will be responsible for fixing the code, updating the documentation, contacting the customer, etc.
- **Resolved** - a bug that has been resolved by a developer has a state of '*resolved*'.
- **Closed** - If a bug is confirmed by QA to have been fixed, it should have a state '*closed*'.
- **Verified** – QA verifies the solution proposed by the developer in order to fix the bug.
- **Reopened** - a '*closed*' bug can be re-opened if it re-surfaced or it is found to be not really fixed.

### 3.3 Mantis

Mantis is the BugTracking System (BTS) selected during WP1 to be the main bugtracker system of the Helios platform.

It is a web based BTS, being released under the terms of the GNU General Public License, available from <http://www.mantisbt.org/>.

#### 3.3.1 Short description of the BTS

According to the project website at <http://www.mantisbt.org/> ...

...Mantis is written in the PHP scripting language and works with MySQL, MS SQL, and PostgreSQL databases and a webserver. It claims support for Microsoft Windows, Mac OS, OS/2, and a variety of Unix operating systems (as part of various Linux distributions), being released under the terms of the GNU General Public License.


The goals are to produce and maintain a lightweight, simple bugtracking system. Additions of complexity/features are modular so that users can be shielded from unwanted clutter. Thus, much of the package has a simple version of a feature along with a more fully developed version.

In the “*core*” package, the aim is to have the most important, most used, most time saving portions of a bugtracking system.

The product is designed to be easily modifiable, customizable, and upgradeable. Anyone with intermediate PHP and MySQL experience should be able to customize Mantis to suit their needs.

#### *Features and Benefits*


- Free under the GNU General Public License
- Easy to install
- Web based

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

- Platform independent
- Supports any platform that runs PHP (Windows, Linux, Mac, Solaris, AS400/i5)
- Multi-DBMS Support - Mantis uses ADODB as an abstraction library to support multiple DBMSes [...]
- Plug-ins support (starting with v1.2.x)
- Export to CSV, Microsoft Excel, Microsoft Word
- Multiple Projects per instance
- Support for Projects, Sub-Projects and Categories
- Users can have a different access level per project
- Multiple languages
- Email notifications
- Search and Filter [...]
- Built-in Reporting (reports / graphs)
- Attachments (can be saved on webserver or in database - can also backup to an FTP account)
- Attachment Auto-Preview
- Issue Change History
- Customizable issue workflow
- Self-Service Reset Password Support
- Ability to extended functionality through hook (custom) functions
- Source Control Integration (SVN and CVS)
- No limit on the number of users, issues, or projects
- Time Tracking
- Ability to view recently visited issues (the last 5 visited issues are visible by default at the top right corner)
- Group Actions: actions can be applied on multiple issues
- Authentication [...]
- Webservice (SOAP API) - MantisConnect project offers client libraries and tools in .NET, Java, and Cocoa. Tools include Eclipse plugin, Ant plugin, notifier.
- Support for mobile devices (MantisWAP)
- Wiki Integration (DokuWiki, MediaWiki, XWiki, TWiki, or none)

The main guidelines that are followed in Mantis are:

- Quick access to "*What I want to do*"

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

- Simple navigation
- Layered complexity
- Consistency
- Scale to browser window size
- Minimal clutter
- Minimal graphics
- No frames
- No animations
- No Javascript

### **3.3.2 Anatomy of a bug**






Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence : LI-15-RAP
Version 1.0 du 16/09/09

<b>Link bug-software:</b>	platform
	OS
	OS version
	product build
	projection
<b>Link bug-persons:</b>	assigned to
	reporter
<b>Link bug-bug</b>	relationships
<b>Bug description</b>	id
	category
	summary
	description
	priority
	severity
	status
	resolution
	steps to reproduce
<b>Miscellaneous</b>	date submitted
	last updated
	reproducibility
	view status
	ETA
	additional information
	tags
	attached tags
	attached files
notes	

Table 2: Mantis bug attributes

According to Mantis manual [7], the bugs (or issues) attributes are :

- Id;
- Category;
- Severity;
- Status;
- Last updated;
- Date Submitted
- Summary

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

- Reproducibility
- Reporter
- View Status
- Assigned To
- Priority
- Resolution
- Status
- Projection
- ETA
- Platform
- OS
- OS Version
- Product Build
- Description
- Steps To Reproduce
- Additional Information
- Tags
- Attach Tags
- Attached Files
- Relationships
- Notes

The types of **severities** are:

1. block - prevents further work/progress from being made
2. crash - crashes the application or OS
3. major - major bug; there is no workaround
4. minor - not trivial, but there is a workaround
5. tweak - needs tweaking
6. text - error in the text
7. trivial - extremely minor issue
8. feature - issue does not describe a problem

The default issue **status** are listed below:

1. new - issue has not been acted upon or assigned to a developer



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence : LI-15-RAP
Version 1.0 du 16/09/09

2. feedback - bug requires more information, the original posters should pay attention
3. acknowledged - bug has been looked at but not confirmed or assigned
4. confirmed - confirmed and reproducible (typically set by an Updater or other Developer)
5. assigned - assigned to a Developer
6. resolved - issue has been provisionally handled, resolution field must be set
7. closed - issue has been definitively handled; should not edit issue without reopening; resolution field must be set.

According to the manual :

Resolution is an issue property, set by the developer when reviewing the issue. It also gives how the issue was fixed, or if not fixed describes why. Some values of resolution are only allowed if the issue status is resolved or closed; these are marked with "\*" below. Note that here, to "*fix*" a defect issue means repair it, but to "*fix*" a feature request means implement it.

Possible **resolution** values are:

- open
- fixed
- reopened: issue Status had been resolved or closed, but is now being reevaluated
- duplicate\*
- works for me: developer couldn't duplicate reported defect\*
- not a problem: product is best without issue being fixed\*
- suspended: will not fix now, but may reevaluate later\*
- won't fix: will not fix for other reasons (e.g. too much work compared to benefit, would cause other problems)
- not fixable


There are four user roles:

- reporter;
- manager;
- developer;
- tester.

The bug life-cycle can be customized. For instance, **Figure 2** explains the work flow for the Mantis application, as proposed by a user<sup>4</sup>.

---

4 Source : <http://my-two-cents.info/software/mantis-bug-tracker-lifecycle-of-a-bug/>

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

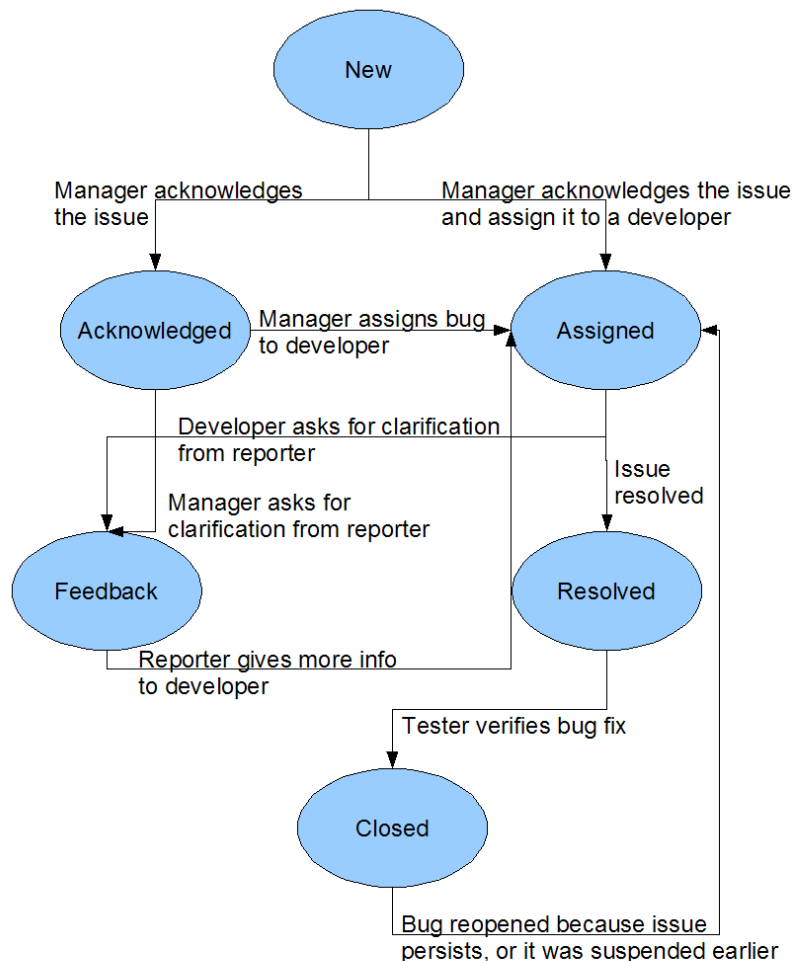


Figure 2: A proposed simple Mantis bugs life-cycle


### 3.4 Debian's Debbugs

**Debbugs** is the Debian project's bug tracking system, which is mainly used by the Debian project to provide <http://bugs.debian.org/>.

It's not a widespread tool, but it has really interesting and original features, and plays a central role in the Open Source ecosystem, managing the hundreds of thousands of bugs reported on Debian packages.

#### 3.4.1 Short description of the BTS

Debbugs consists of a set of scripts which maintain a database of problem reports. Most of the

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

source code is written in Perl. It is a free software and it's licensed under the GNU General Public License.

According to the program's old web page<sup>5</sup>, the main features of Debbugs are:

- All input and manipulation of reports is done by email, thus developers do not need on-line web access or accounts on the host system.
- Outstanding, recently closed and other listings of reports are made available via a webserver, and by an email request bot.
- Each report has a separate e-mail address for submission of additional information.
- Web listings of bugs can be mirrored by e-mail or HTTP.
- Core functions do not require CGI scripts.
- The scripts have been parameterized so that they can be used for other projects besides Debian.
- The system runs on Unix-like operating systems such as Unix and Linux.

There are two ways to report a bug in Debian:

- by running from the command line the program **reportbug** (this is the recommended way to report bugs)
- by using e-mail

This makes it a bit different from other web-based bug-tracking systems that use web forms for doing so, most of the time.

Anthony Towns , in “*debbugs – Tips, Tricks and Hacks*”[11], expresses some of the advantages of Debbugs are:

- Familiar interface for Debian users – thus allowing developers to interact with the system using email, and provide all the data to anyone with a web browser.
- Package based – it handles quickly and easily the separating bug reports by package.
- Scalability – can cope with a large number of bug reports.
- Quick response – it provides immediate feedback on the state of bugs, and handle requested changes promptly.
- Reliability – it continues working at all times, even as new features are trialled.
- Public – it provides a permanent public record of discussions, and allows the members of the wider Debian community to participate in those discussions with absolutely minimal effort.

---

<sup>5</sup> <http://www.chiark.greenend.org.uk/~ian/debbugs/> whereas one would nowadays refer to <http://wiki.debian.org/BTS> instead



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence : LI-15-RAP
Version 1.0 du 16/09/09

### 3.4.2 Anatomy of a bug

<b>Link bug-software:</b>	package
	found_version
	fixed_version
	fixed_date
	source
<b>Link bug-persons:</b>	originator
	owner
<b>Link bug-bug</b>	forwarded
	mergedwith
	blocks
	blockedby
	affects
<b>Bug description</b>	id
	fixed
	found
	pending
	done
	subject
	severity
	summary
	bug_number
<b>Miscellaneous</b>	date
	tags
	unarchived
	log_modified
	location
	archived
	msgid
	keywords

Table 3: Debbugs bug attributes

The attributes describing a bug in Debbugs are:

- submitter
- date
- subject
- message\_id



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie

Référence : LI-15-RAP

Version 1.0 du 16/09/09

- package
- tags
- done
- forwarded\_to
- merged\_with
- severity
- owner
- found\_in
- found\_date
- fixed\_in
- fixed\_date
- blocks
- blocked\_by
- unarchived
- summary
- affects

The bug lifecycle is XXX TO BE DONE XXX

## 3.5 Launchpad's bugtracker

**Launchpad** is a web-based application and website (<http://launchpad.net/>) supporting software development, particularly for free software. Launchpad is developed and maintained by Canonical Ltd and it is written in Python. Launchpad's code has been released in an open way in July 2009.

### 3.5.1 Short description of the BTS

Launchpad is currently heavily used in the development of the Ubuntu GNU/Linux distribution. But it's also used by many projects hosted on launchpad, allowing easy management of links between a bug in such a project and the counterpart in its Ubuntu packages, as expressed on the site :

When projects share free software code they share its bugs. Inevitably, the same bug is reported and discussed in different places. With Launchpad it is possible to monitor on one page how the bug affects different communities, whether they're tracking the bug in Launchpad, Bugzilla, Trac, Sourceforge, or elsewhere. It is possible even to share a comment history across project-boundaries, bringing everyone together to find a fix.



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence : LI-15-RAP
Version 1.0 du 16/09/09

### 3.5.2 Anatomy of a bug


<b>Link bug-software:</b>	component
	milestone
<b>Link bug-persons:</b>	reporter
	subscribers
	assignee
<b>Link bug-bug</b>	duplicate-of
	duplicates
<b>Bug description</b>	bug
	title
	status
	importance
<b>Miscellaneous</b>	date-reported
	date-updated
	tags
	task
	date-created
	date-left-new
	date-confirmed

**Launchpad Bug Model**

The fields that describe a bug are:

- bug
- title
- date-reported
- date-updated
- reporter
- duplicate-of
- duplicates
- attachments
- tags
- subscribers
- task



	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

- status
- date-created
- date-left-new
- date-confirmed
- importance
- component
- assignee
- milestone

According to the documentation, the bug tracker can take seven main bug statuses :

Available to everyone


- *New*
- *Incomplete*: the bug report is incomplete and needs more information before it can be triaged. Bugs in this state are considered for expiry.
- *Invalid*: the report describes the software's normal behavior, or is unsuitable for any other reason.
- *Confirmed*: a member of the community other than the original reporter believes that this report describes a genuine bug in enough detail that a developer could start work on a fix.
- *In Progress*: a developer has taken responsibility to fix the bug and has begun work.
- *Fix Committed*: a developer has committed his/her fix to the project's codebase.
- *Fix Released*: a new version of the software, featuring the bug fix, has been released.

Only available to the bug supervisor

- *Triaged*: the bug supervisor considers that the bug report contains all information a developer needs to start work on a fix.
- *Won't Fix*: this is acknowledged as a genuine bug but the project has no plans to fix it.

The severity of a bug in Launchpad is described by the field *importance*. The possible values are:

- undecided
- wishlist
- low
- medium
- high
- critical

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

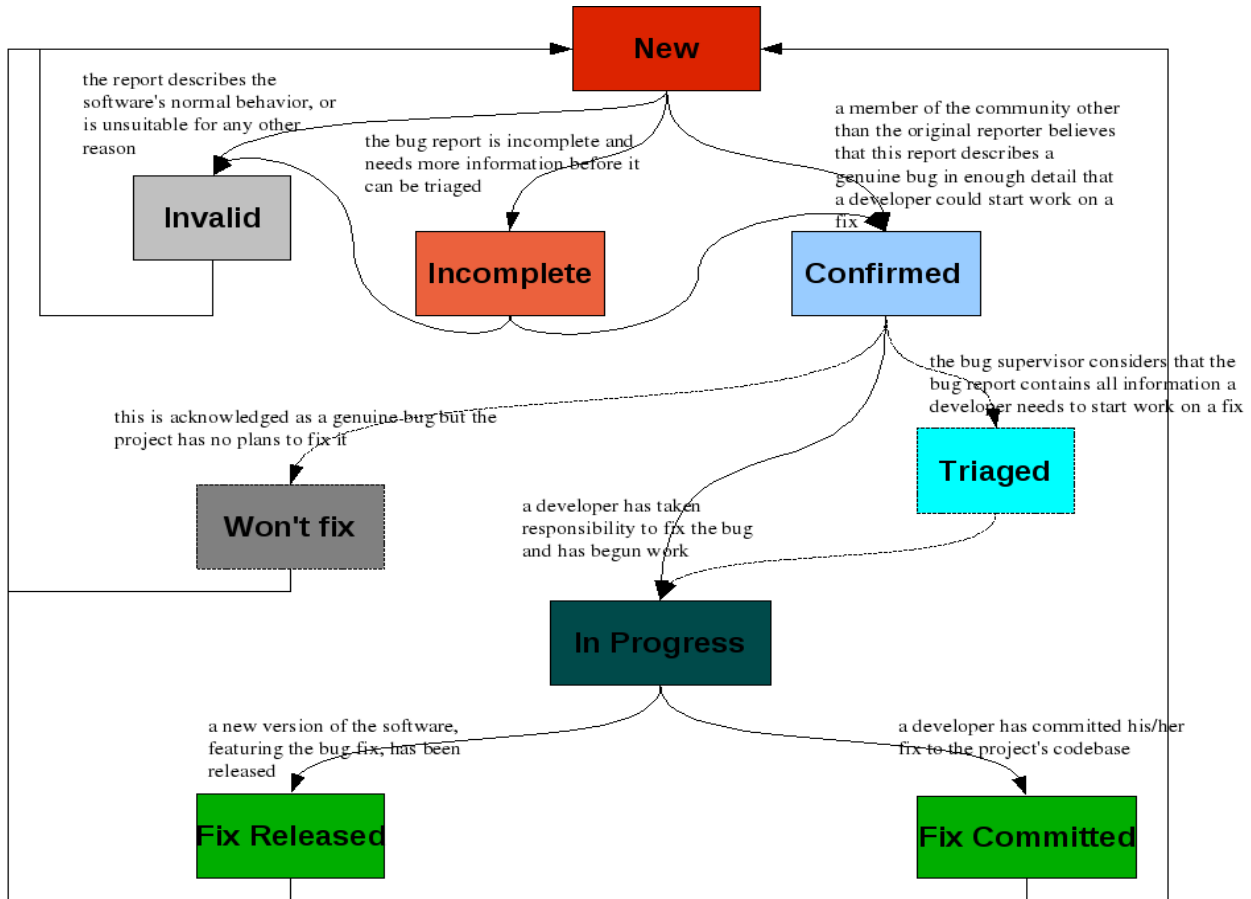


Figure 3: Launchpad bug lifecycle

### 3.6 Gforge / FusionForge's tracker


**FusionForge** is the new name of the FLOSS version of the GForge application (started recently in 2009). It is a web based application, written in PHP and under the GNU General Public License (GPL), available from <http://fusionforge.org/>.

FusionForge helps managing the entire development life cycle by using tools for team collaboration, like message forums and mailing lists; tools to create and control access to Source Code Management repositories like CVS and Subversion. FusionForge automatically creates a repository and controls access to it depending on the role settings of the project.

#### 3.6.1 Short description of the BTS

Among lots of other features, FusionForge includes a configurable **Issue tracking** with "unlimited" numbers of categories, text fields, etc.

By default, on project creation, FusionForge uses the Trackertool tool as an issue manager, but it

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

may be used for other needs with completely different metamodel for tickets/artefact properties.

### 3.6.2 Anatomy of a bug

<b>Link bug-software:</b>	component
	product
	OS
	version
<b>Link bug-persons:</b>	hardware
	assignend_to
<b>Link bug-bug</b>	submitted_by
	url
<b>Bug description</b>	duplicates
	artifact_id
	group_artifact_id
	status
	bug_resolution
	priority
	severity
<b>Miscellaneous</b>	summary
	open_date
	close_date
	last_modified_date
	details


Table 4: FusionForge Bug attributes

The tracker's main element is called artifact and can belong to categories in four distinct groups:

- bugs
- support
- patches
- feature request

The fields that defines an artifact as a bug are:

- artifact\_id
- group\_artifact\_id
- status (open, closed)

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

- priority
- hardware
- product
- os
- component
- version
- bug\_resolution ( *accepted as bug, fixed, won't fix, invalid, awaiting response, works for me* )
- url
- summary
- open\_date
- close\_date
- last\_modified\_date
- severity (blocker, critical, major, normal, minor, trivial, enhancement)
- assign\_to
- submitted\_by
- details

Life-cycle : XXXTODOXXX

### 3.7 Trac's bugtracker

According to Trac's website<sup>6</sup>, it's ...

... a lightweight project management tool that is implemented as a web-based application. It is written in the Python programming language and it's used for managing software developments, being flexible enough to be suited for different types of projects. Because it's an open source project, it's code can be modified by writing new plugins.


Trac runs on any system which supports Python and the depending modules (various Linux distributions, Mac OS X, FreeBSD, NetBSD and MS Windows).

Starting with version 0.9, Trac was released under the modified BSD license, but prior of that it was released under the GNU General Public License (GPL).

It provides an interface to Subversion (or other version control systems), an integrated Wiki and convenient reporting facilities.

---

<sup>6</sup> <http://trac.edgewall.org/>

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

### 3.7.1 Short description of the BTS

The integration of Wiki allows Trac to make wiki markup in ticket descriptions and commit messages, creating links and seamless references between bugs, tasks, changesets, files and wiki pages. A timeline shows all current and past project events in chronological order, while the roadmap lists the upcoming milestones.

Another important feature is TracLinks which can be used in ticket descriptions and comments to refer to other issues, changesets or files, to make the ticket more specific and easier to understand.

### 3.7.2 Anatomy of a bug


<b>Link bug-software:</b>	component
	version
	milestone
<b>Link bug-persons:</b>	assignend to/owner
	reporter
	cc
<b>Link bug-bug</b>	url
	duplicates
<b>Bug description</b>	id
	type
	status
	resolution
	priority
	severity
	summary
description	
<b>Miscellaneous</b>	time
	changetime
	attachment
	keywords

*Table 5: Trac's bug attributes*

The central project management element of Trac is the ticket which can take various roles like:

- defect,
- enhancement,
- task.

A ticket is assigned to a person who must resolve it or reassign it to someone else. The tickets can be edited, annotated, assigned, prioritized and discussed at any time. All the changes and comments to the ticket are logged as a part of the ticket itself.

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

The fields that describes a Trac ticket are:

- id
- type
- time
- changetime
- severity (*blocker, critical, major, normal, minor, trivial*)
- priority (*highes, high, normal, low, lowest*)
- component
- owner/assigned to
- reporter
- cc
- version
- milestone
- status (*'new', 'assigned', 'reopened', 'closed'*)
- resolution (*'fixed', 'invalid', 'wontfix', 'duplicate', 'worksforme'*)
- summary
- description
- keywords
- attachment

Starting with the version 0.11, when a new environment is created, a default workflow is configured in file trac.ini. This workflow is the basic workflow (described in basic-workflow.ini) and graphically, it looks like this:

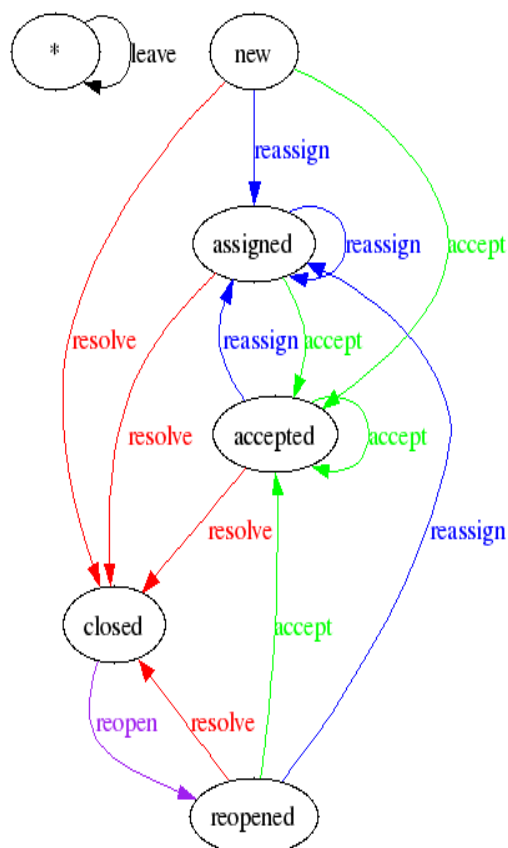


Figure 4: Trac's bugs lifecycle



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence : LI-15-RAP
Version 1.0 du 16/09/09

## 4 Bug models

This section will describe some of the existing models which have been elaborated to model bugs, that existed prior to the start of this study. Then it will introduce the Helios\_BT model that was designed to complement them.

We focus in particular on Semantic-Web ontologies that have been proposed to represent facts about bugs, and their representation in standard formats such as RDF. These ontologies will generally be described using OWL.

The Semantic Web standards are, one of the the best technologies currently available that allow to deliver the promises of interoperability between many different bugtracking systems (and other applications) that all use different languages, databases and architectures.

### 4.1 Baetle project

**Baetle** stands for **Bug And Enhancement Tracking Language**. The *baetle project*<sup>7</sup>, started in 2007 aims at producing an “ontology to describe information kept in bug databases”.

There is currently a loose community of interest around the baetle project trying to improve the model in an open way.

Although an ontology (cf. Ontology 1) was drafted it seems it hasn't been used for real applications .

<b>Title</b>	Baetle (Bug And Enhancement Tracking Language)
<b>Author</b>	Henry Story and others
<b>Version considered</b>	Unreleased draft
<b>URL</b>	<a href="http://code.google.com/p/baetle/">http://code.google.com/p/baetle/</a>
<b>Schema</b>	<a href="http://code.google.com/p/baetle/source/browse/ns/">http://code.google.com/p/baetle/source/browse/ns/</a>
Ontology 1: Baetle project's	

Baetle's model was heavily inspired by the Bugzilla bug attributes.

EvoOnt BOM (see 4.1) is currently considered by some participants of the Baetle project to be somehow the most concrete and validated instantiation of the Baetle initiative.


### 4.1 EvoOnt BOM

EvoOnt BOM is in our opinion the most elaborated ontology available prior to our work to model

---

<sup>7</sup> <http://code.google.com/p/baetle/>



	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

bugs facts.

#### 4.1.1 EvoOnt ontologies

EvoOnt stands for “A Software Evolution Ontology”. According to its description<sup>8</sup> :

*“EvoOnt is a set of software ontologies and data exchange format based on OWL. It provides the means to store all elements necessary for software analyses including the software design itself as well as its release and bug-tracking information.”*

There are 3 main components in EvoOnt :

- Version Ontology Model (VOM)
- Software Ontology Model (SOM)
- Bug Ontology Model (BOM)

---

8 <http://www.ifi.uzh.ch/ddis/evo/> (visited 2009/03/26)

Figure 5 illustrates the main entities and relations between them<sup>9</sup>.

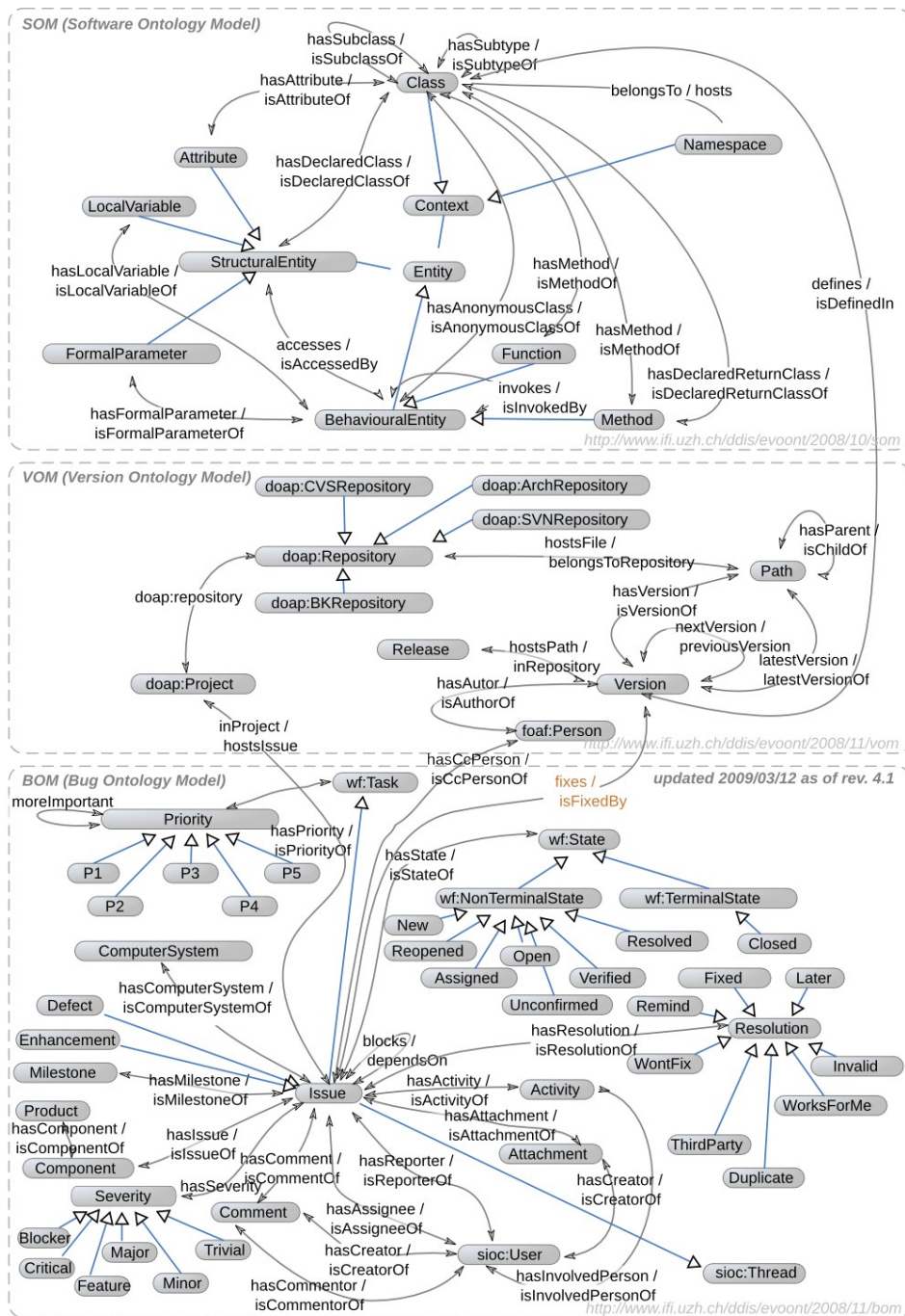


Figure 5: EvoOnt ontology classes and relations

Of these three models, the most interesting one for our concerns is **BOM** (Bug ontology Model).

<sup>9</sup> Source : <http://code.google.com/p/baetle/source/browse/evoont/trunk/> (improved in the frame of the present study)

### 4.1.2 BOM (Bug Ontology Model)

The reference version considered at the start of this study is rev. 4.1, which is available online from its original authors' site (see Ontology 2).

The central class/entity in the BOM ontology is the **Issue**, which can be refined in Defect (bug) or Enhancement (feature request).

Figure 6 shows the relations between Issues and other entities of the ontology, some being already standardised in other ontologies (for persons / accounts like sioc:User, mainly).

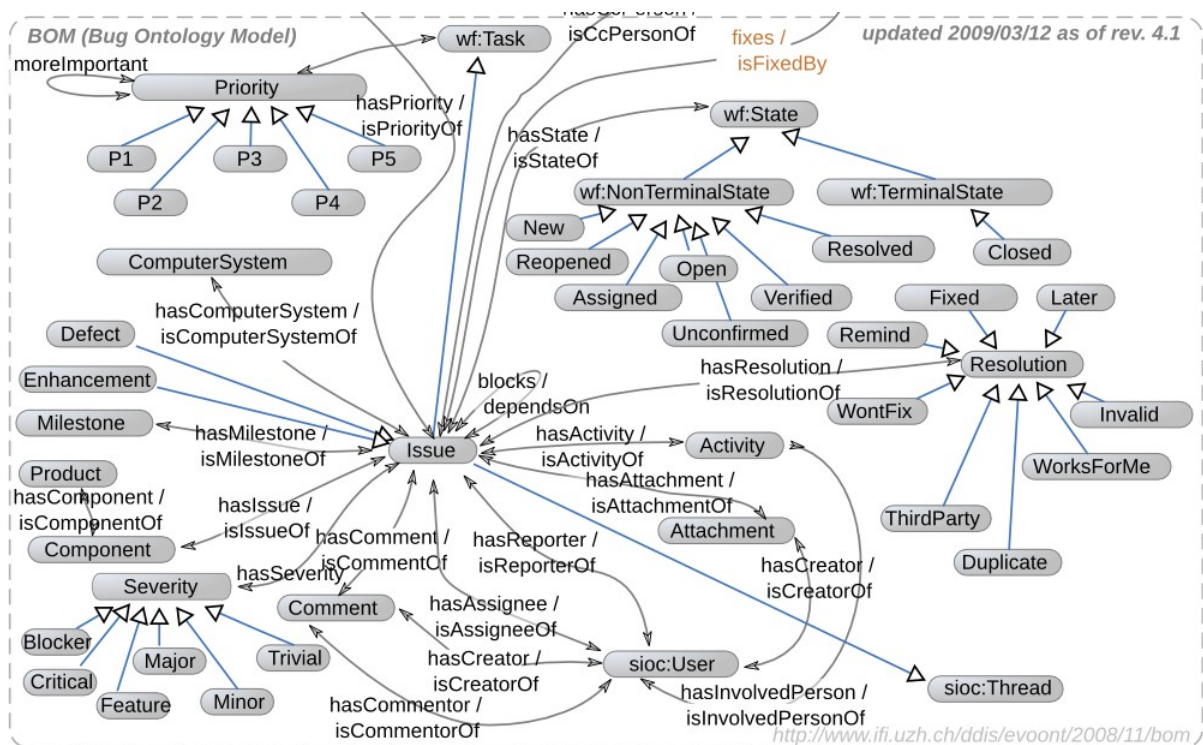



Figure 6: EvoOnt BOM's classes and relations

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

<b>Title</b>	EvoOnt's Bug Ontology Model
<b>Author</b>	DDIS Uni Zurich – Jonas Tappelet
<b>Version considered</b>	4.1
<b>URL</b>	<a href="http://www.ifi.uzh.ch/ddis/evo/">http://www.ifi.uzh.ch/ddis/evo/</a>
<b>Schema</b>	<a href="http://www.ifi.uzh.ch/ddis/evoont/2008/11/bom/bom.owl">http://www.ifi.uzh.ch/ddis/evoont/2008/11/bom/bom.owl</a>
<i>Ontology 2: EvoOnt BOM (Bug Ontology Model)</i>	

Although BOM was developed for research purposes, initial tests tend to indicate that it would fit the needs for use in implementation scenarios on bug management applications too, hence in the context of the Helios project.

Note that we have proposed a few minor modifications (syntactic mainly) and are trying to collaborate with its authors to propose their integration in a next version. In the meantime, we publish the current version of such a modified version in the Baetle project's repository<sup>10</sup>.

### 4.1.3 BOM in details


In addition to the entities visible in Figure 6, BOM also defines lots of entities, attributes and relations to characterize the bugs, which fit particularly well with the surveyed bugtrackers (see previous section).

A detailed auto-generated documentation of EvoOnt BOM is available online at : <http://www.ifi.uzh.ch/ddis/evoont/2008/11/bom/doc/index.html>

We have identified some limitations in BOM, though :

- lack of modeling of relations between different bugs (duplicate bugs, forwarded-to, etc.),
- only one generic list of “constant” values for attributes like state, severities, priorities and other such enumerated values : this doesn't fit all bugtrackers (BOM mainly models Bugzilla's lifecycle),
- weak model of links between bugs and software packages, and in particular versions of these (the links with SOM & VOM mainly model links with source code), for instance in order to track bugs between debian packages,

<sup>10</sup> At the time of writing these are available in <http://code.google.com/p/baetle/source/browse/evoont/trunk/bom>

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

## 4.2 Our proposition: Helios\_BT to complement BOM

We intend to complement EvoOnt BOM in order to add the necessary attributes and entities that are required for the needs of the Helios WP3, and in order to address the studied bugtrackers specifics.

The diagram in Figure 7 represents an overview of the core entities that will be modeled in addition to the EvoOnt BOM existing entities and relationships.

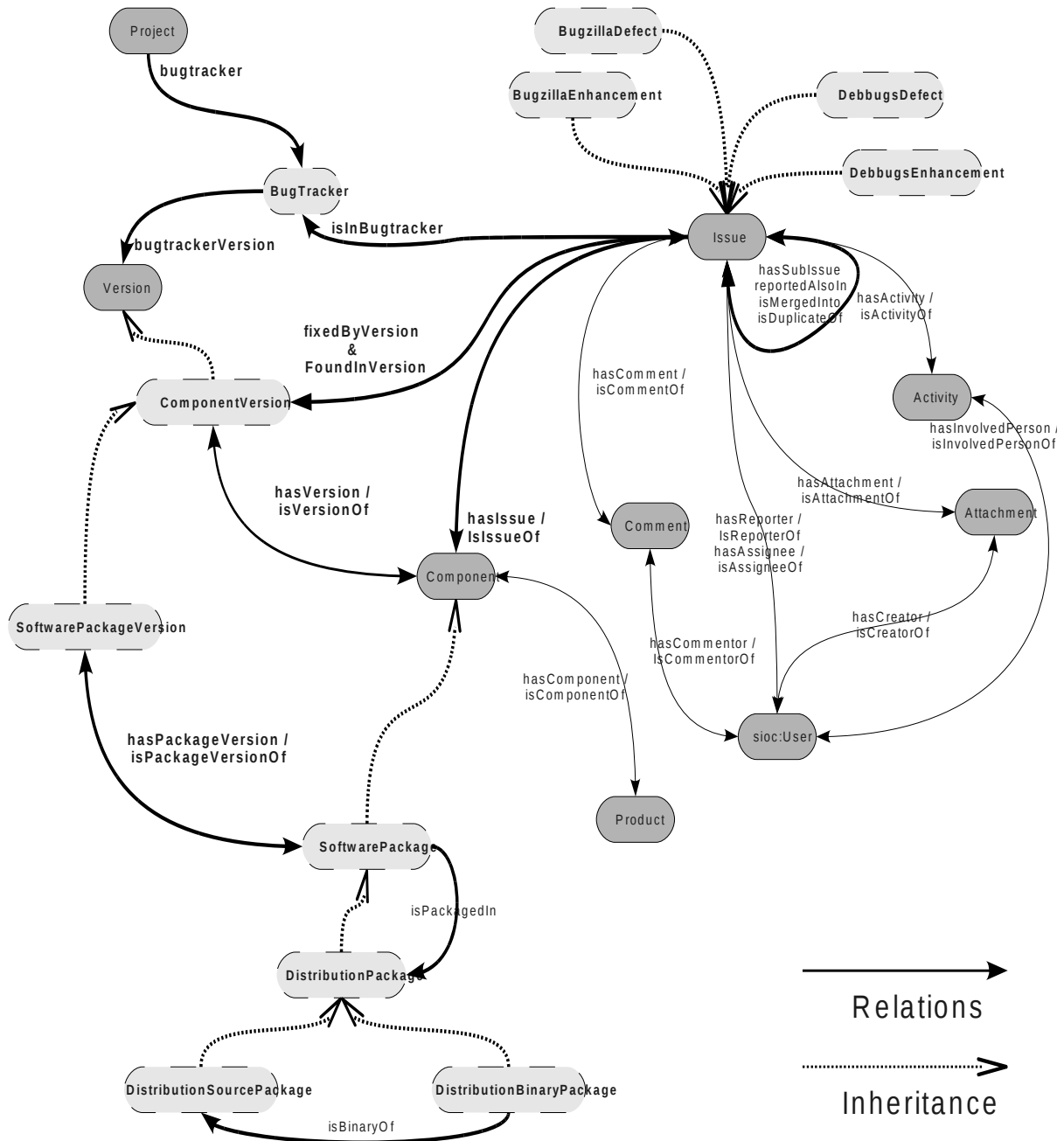



Figure 7: Overview of new Helios\_BT entities and relations

In that figure, entities with dashed borders represent new additions in Helios\_BT compared to BOM, and bold lines represent new relations.

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

Helios\_BT (Helios Bug Tracking) is the prefix we propose for an ontology filling such a need. It's meant to be available at the stable URL : [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/helios\\_bt.owl](http://picoforge.int-evry.fr/projects/svn/helios_wp3/helios_bt.owl) as OWL.

Note that the above URL is a redirector, and versioned URIs like : [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/09/helios\\_bt.owl](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/09/helios_bt.owl) are used in implementation.

The reference documentation web page for this ontology is available at : [https://picoforge.int-evry.fr/cgi-bin/twiki/view/Helios\\_wp3/Web/HeliosBtOntology](https://picoforge.int-evry.fr/cgi-bin/twiki/view/Helios_wp3/Web/HeliosBtOntology) (a work-in-progress wiki page, updated whenever necessary).

See Annexes of the present document for the full auto-documentation of the ontology for complete details<sup>11</sup>.

Helios\_BT provides the following additions or differences to EvoOnt BOM :

- In EvoOnt's VOM and BOM ontologies, bugs are linked to versions of source files that have been committed in the source repositories in order to fix them. Such approach doesn't suit the needs in the context of the Helios WP3, where we intend to track instead the links between released versions of components (software packages) that are reported as including a fix for a bug (in the changelog).  
See the proposed SoftwarePackageVersion entity in Figure 7 and the foundInVersion and fixedByVersion links that we propose to add in Helios\_BT.
- for each bugtracker, a set of fixed values (classes) for severity, priority, etc. : semantically, they are significant only for one particular flavour of bugtracker. Equivalences between values of different bugtrackers' values (i.e. Bug lifecycle Workflow equivalences) won't be discussed in Helios for the moment. It is left to further studies to provide further refinements through OWL reasoning, for instance.
- several relationships are proposed to model links between bugs (duplicates, also present in other bugtrackers, etc.), as illustrated in Figure 7 by the relations isDuplicateOf / isMergedInto, etc.

Other elements include :


- the description of the bugtracker of a DOAP project (class BugTracker in Figure 7)

#### 4.2.1 Bug attributes modeling

We propose to represent most of the the bug properties with EvoOnt BOM and Helios\_BT, at least in order to fulfill the needs of applications to be developped as part of Helios. Not all bugtrackers's

<sup>11</sup> Only version 07/2009, slightly outdated at the time of last revision of this document, has been included, the reference up-to-date version being on-line.



	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

specific models may need to be described in full details at the moment. Should converters from bugtracker be considered, then complementary ontology extensions could be developed.

The bug's discussions can be represented with SIOC-related discussion threads, as proposed by EvoOnt, but not much work was conducted on these elements at the time of writing.

#### 4.2.2 Links between Bugs and Software / Distribution / Package / Versions

This is not yet completely covered by the version of the ontology documented in the annex and will be complemented in later phases to fulfill the needs of other workpackages of Helios (integration of the bugtracker with the rest of the platform).

#### 4.2.3 Links between a Bug and other Bugs

This has already been addressed in Helios\_BT, and will be tested on further developments on the btlink tool in WP3.

#### 4.2.4 Links between Bugs and Persons

The model in EvoOnt is already describing such links, as illustrated in Figure 8 by linking with FOAF and SIOC entities which are already standards in Semantic Web applications.

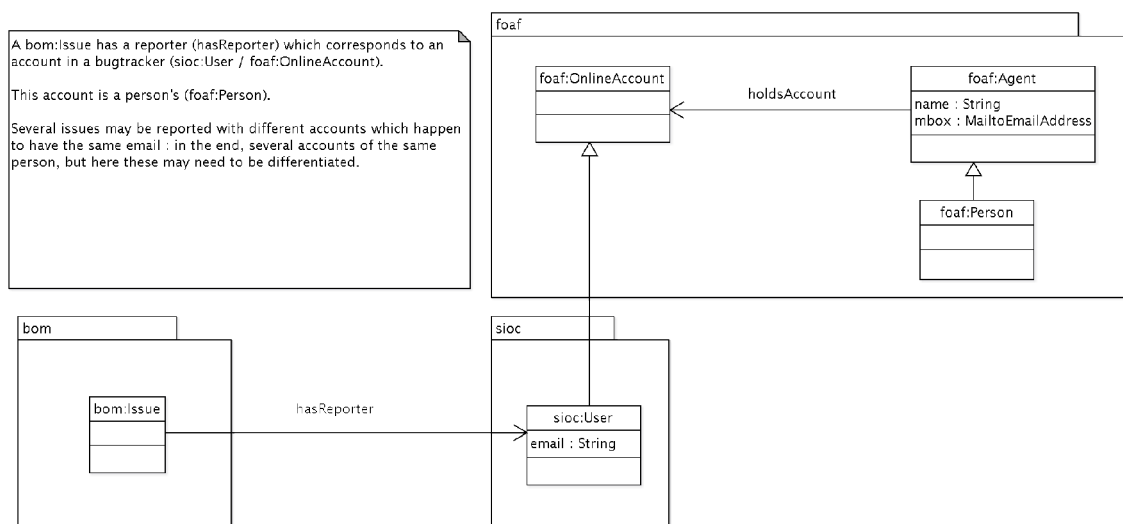



Figure 8: Links between bugs and persons

This has been tested in order to construct social network representations out of bugs databases<sup>12</sup>.

12 [http://www-public.it-sudparis.eu/~dang\\_qua/pmwiki/pmwiki.php?n=Blog.DebianSocialNetwork](http://www-public.it-sudparis.eu/~dang_qua/pmwiki/pmwiki.php?n=Blog.DebianSocialNetwork)



	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

### 4.3 Related ontologies and standards

Some widely used ontologies are used, to describe facts related to entities linked to the bugs in the Helios\_BT ontology (or its underlying BOM base) :

- FOAF (Friend Of A Friend) : describe relations between people
- DOAP (Description Of A Project) : describes software development projects
- SIOC (Semantically-Interlinked Online Communities) : describes persons, accounts on online services and discussions

The mixing of our novel ontology and these existing accepted standards will facilitate the construction of rich and standardized interfaces for semantics-aware bugtracking tools.



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence : LI-15-RAP
Version 1.0 du 16/09/09

## 5 Standardization

This sections deals with standardaization effort of the bug models.

We seek for the establishment of a standard for bugs description and interchange between applications, by developping a generic model that suits most needs of today, and which can be extended (thanks to the inherent extensibility of RDF) for tomorrows needs.

We'll develop some tools in Helios based on such a standard, and do our best to disseminate our approach and results, but of course, only the future will tell if it may become a standard in this field.

### 5.1 Demonstrators

Several demonstrators are developped to validate the use of these ontologies :

- Triplify<sup>13</sup>-based web applications to publish bug facts ou of the databases of some of the bugtrackers we studied:
  - bugzilla
  - mantis
  - debbugs (through Debian's UDD database)

They can be installed to provide RDF documents out of bugs databases, and navigate the Semantic Web graph, for instance with appropriate Firefox plugins.


### 5.2 Dissemination efforts

In addition to the Baetle community (see 4.1 ), we took contacts with various communities to promote our efforts, and share ideas for implementation of bug models and standards of interchange :

- bts-link developpers
- UDD maintainers of the Debian project
- Bugzilla developpers
- participation to the triplify challenge (see “Bugtrackers triplification”)
- presentation at OSS 2009 / WOPDASD 2009 (see “Weaving a Semantic Web across OSS repositories: a spotlight on bts-link, UDD, SWIM”)
- presentation at RMLL/LSM 2009
- many weblog posts by Olivier Berger


---

13 <http://triplify.org>

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09


### **5.3 Future efforts**

Implementation of tools in WP3 that can work on the proposed model and ontology will demonstrate for efficiently the validity of the approach and will help disseminate its use.

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

## 6 References

- [1] Matthew B. Doar - “Common Problems with Bug Trackers”, September 2006, ACCU ([http://www.accu-usa.org/Slides/Common\\_Problems\\_with\\_Bug\\_Trackers.pdf](http://www.accu-usa.org/Slides/Common_Problems_with_Bug_Trackers.pdf))
- [2] [http://www.ozibug.com/www/bug\\_tracking.html](http://www.ozibug.com/www/bug_tracking.html)
- [3] Davor Cubranic, Gail C. Murphy - “*Automatic bug triage using text categorization*”
- [6] The Bugzilla Guide - The Bugzilla Team (as of documentation of version 3.1.3 Development Release, Published 2008-02-01)
- [7] Mantis Manual v.1.0.0, Last Modified: April 22, 2005
- [11] Anthony Towns , “*debbugs – Tips, Tricks and Hacks*”, June 2005 (<http://code.erisian.com.au/Wiki/debbugs>, and <http://debconf5.debconf.org/comas/general/proposals/47.html>)
- [12] <http://www.debian.org/>
- [13] <http://wiki.debian.org/DebbugsSoapInterface>
- [14] [https://developer.mozilla.org/en/Bug\\_writing\\_guidelines](https://developer.mozilla.org/en/Bug_writing_guidelines)
- [15] <http://www.ifi.uzh.ch/ddis/evo/>

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

## **DEUXIÈME PARTIE :**

# **Description des interfaces sous formes de Web Services du suivi d'anomalie**



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence : LI-15-RAP
Version 1.0 du 16/09/09

## 7 Définition du besoin

En introduction de ce chapitre, nous rappelons la définition des tâches spécifiées dans l'annexe technique pour le WP3 « Traçabilité des anomalies entre plateformes de développement ».

Cette définition est ensuite complétée pour présenter l'état des besoins en matière de gestion de bugs et une description globale du périmètre des applications à développer dans le cadre du WP3.

### 7.1 Rappels des éléments définis dans l'annexe technique

L'annexe technique mentionne (nous soulignons en gras les éléments essentiels) :

*« Une application construite à base de composants Open Source est le résultat du travail de plusieurs groupes qui utilisent chacun leurs propres environnements de développement de type forge (développeurs des briques de base, développeurs du logiciel et développeurs de la ou des distributions). Ceci est vrai pour chaque version de l'application. Dans ce contexte il est difficile de **savoir si un bug (anomalie) identifié par une version remonte suffisamment loin puis redescend dans toutes les versions.** De même il est difficile **d'assurer qu'une correction de bug dans une bibliothèque est reportée dans tous les logiciels qui l'utilisent.** Les processus de remontée de bugs sont, à ce jour, manuels. Ce WP s'attachera à proposer des mécanismes d'interaction associés à ces traitements et à **permettre la traçabilité de ces interactions pour faciliter le suivi du traitement de bug** par l'ensemble des groupes concernés. Les mécanismes proposés seront intégrés dans la plateforme d'industrialisation pour faciliter les **interactions entre les groupes travaillant autour des mêmes logiciels.***

*Objectif :*

*L'objectif de ce WP est de **développer un outil de synchronisation entre bug-trackers** (outils de suivi d'anomalies) en prenant en compte les bug-trackers les plus couramment utilisés et Open Source, à savoir Bugzilla, Mantis, Trac entre autres.*

[...]

#### Tâche 3.1 : Étude et modélisation

*Cette étude proposera non seulement un modèle générique d'anomalie, mais aussi des interfaces sous forme de Web Services pour automatiser l'enchaînement des traitements associés aux anomalies. La description des inter-relations entre briques logicielles telle qu'on peut la trouver dans les gestionnaires de paquet logiciel utilisés dans les distributions libres (rpm, et deb) sera utilisée pour **faciliter la recherche des relations entre les différents niveaux auxquels un bug peut s'exprimer.** Cette étude proposera des scénarios pour analyser la **circulation de l'information relative aux traitements des erreurs** dans le cas de **circulation ascendante** (d'une brique utilisatrice vers une brique de base), **descendante** (d'une brique de base vers une brique utilisatrice), ou **horizontale** (entre briques de même niveau qui utilisent les mêmes briques de base). Ce modèle peut s'appuyer sur des études existantes comme Beatle ou les formats utilisés par des logiciels tels que*



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie

Référence : LI-15-RAP

Version 1.0 du 16/09/09

### *ScmBug.*

*L'un des livrables de cette tâche correspond à une proposition de norme de description d'un bug, ainsi que des interfaces d'interaction avec un gestionnaire de bug. Certains résultats du projet européen Edos pourront être réutilisés.*

*Livrable : LI-15-RAP (dû à T0+8) Spécification de modèle générique de suivi d'anomalie et une description des interfaces sous formes de WebServices du suivi d'anomalie*

### *Tâche 3.2 : Développement d'un mécanisme de synchronisation entre bug-trackers.*

*Ce mécanisme devrait **permettre à un bug-tracker de se synchroniser avec un second bug-tracker sur certaines anomalies bien identifiées.** Les bug-trackers concernés par cet outil sont les suivants :*

- Bugzilla*
- Mantis*
- Trac*
- Bug-tracker LibreSource*

*Les interfaces définies dans les tâches seront réalisées dans au moins deux systèmes de suivi de bugs utilisés par les communautés de développement des logiciels libres en particulier ceux qui sont associés avec des plates-formes de développement de logiciels telles que LibreSource et Gforge. L'interconnexion entre ces deux systèmes sera la preuve de la faisabilité du mécanisme. Les connecteurs construits selon ce modèle et intégrés aux logiciels libres utilisés seront développés en collaboration avec la communauté correspondante afin d'en faire profiter le plus grand nombre rapidement. Les connecteurs associés à des systèmes de Wiki seront aussi réalisés et diffusés.*

*Livrable : LI-16-LOG (dû à T0+19) Logiciel correspondant à une implémentation du mécanisme de synchronisation dans deux bug-trackers. Le système sur lequel cette implémentation sera réalisée dépend des choix du WP1.*

### *Tâche 3.3 : Interaction avec les autres WPs*

*Réalisation d'une interface de **gestion des bugs au sein du portail d'industrialisation** (voir WP4). Cette tâche doit notamment permettre de sélectionner un composant pour **transférer le bug à la communauté concernée et marquer le bug comme transféré.** Cette anomalie sera bien évidemment **transférée puis synchronisée** grâce à l'outil développé dans la tâche 3.2.*


*Cette tâche s'intéresse également aux possibles déclinaisons du modèle de suivi de bug dans le cadre de la gestion de la qualité documentaire. En particulier, elle investiguera les possibles **interactions entre des systèmes de type Wiki et des systèmes de type suivi des erreurs.***

*Livrable :*

- LI-17-LOG (dû à T0+21) Logiciel correspondant à l'intégration des résultats de la tâche 3.3 dans le WP4*
- 

### *Livrables finaux :*

- LI-18-RAP (dû à T0+21) Spécification de modèle générique d'anomalie reprenant le livrable de la tâche 3.1 définie ci-après,*

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

- *LI-19-RAP (dû à T0+21) Documentation des interfaces d'un service de suivi des anomalies reprenant le livrable de la tâche 3.1 définie ci-après,*
- *LI-20-LOG (dû à T0+21) Logicielle sous forme de code source correspondant à l'implémentation d'un mécanisme de synchronisation entre bug-trackers prenant en compte deux bug-trackers reprenant le livrable de la tâche 3.2 définie ci-après,*
- *LI-21-LOG (dû à T0+21) Logiciel sous forme de code source correspondant à l'intégration d'un service de suivi des anomalies dans le WP4 reprenant le livrable de la tâche 3.3 définie ci-après, »*

## 7.2 Compléments à l'issue d'une première phase d'étude

À l'issue de la tâche 3.1 de spécification, il nous apparaît nécessaire de clarifier certains points et de reformuler les objectifs de l'étude, afin de mieux répondre aux problématiques rencontrées. Cela permettra de définir précisément le périmètre des développements à réaliser dans le cadre du WP3, ainsi que les développements complémentaires envisageables.

La notion de synchronisation entre bugtrackers nécessite d'être explicitée. Nous l'entendons comme étant l'ensemble des mécanismes qui peuvent être mis à disposition des « utilisateurs » (utilisateurs de logiciels ou développeurs/mainteneurs), dans le cadre des processus de gestion (de rapports) d'anomalies, afin de leur faciliter la tâche de suivi des liens existants entre différents rapports de bugs, et des mises-à-jour éventuelles nécessaires de l'état de ces rapports dans les bugtrackers.

Ce type de besoin peut exister par rapport à différents processus, et nous ne prétendons pas couvrir tous les cas possibles dans le cadre du projet HELIOS.

Nous nous attacherons donc à répondre à certains besoins précis identifiés plus bas, et à fournir une technologie permettant une extension facilitée pour répondre à d'autres besoins, au-delà du projet ou des fonctions envisagées ici.


Le cœur de cette technologie consistera en **un système de suivi de bugs global**. Ce système permettra de modéliser, au sein d'un même référentiel, un ensemble de propriétés relatives à des rapports de bugs physiquement présents dans leurs propres bugtrackers. Il constituera la brique de base pour faire fonctionner **différentes applications** de suivi des évolutions de ces propriétés, mais aussi des liens existants entre ces bugs. Ces applications seront spécialisées, et répondront chacune à des objectifs particuliers.

Nous décrivons ci-après un ensemble de telles applications, mais seules certaines d'entre-elles seront réalisées dans le cadre du projet.

## 7.3 Contextes d'utilisation

Nous listons différents contextes d'utilisation envisagés pour les technologies développées dans le cadre défini ci-dessus.



	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

Certaines applications seront développées dans le cadre du projet car elles s'articulent particulièrement bien avec les autres outils développés dans celui-ci. Ces applications sont relatives au contexte d'Application Lifecycle Management, ou aux processus de maintenance de logiciels OpenSource. D'autres applications sont listées ci-dessous à seule fin d'illustrer le potentiel des technologies développées.

### 7.3.1 Suivi générique de bugs saisis dans des bugtrackers externes

Au sein d'un projet de développement particulier, doté d'un bugtracker configuré pour ses besoins, le cycle de vie des rapports de bugs est imposé par le couple bugtrackers / projets. Seuls des utilisateurs autorisés ont les prérogatives pour gérer les évolutions de certaines propriétés des rapports de bugs (statut notamment : ouvert / fermé, etc.).

Par contre les soumissionnaires des rapports de bugs extérieurs au projet peuvent avoir leurs propres processus de gestion des rapports d'anomalies constatées, sans qu'ils puissent nécessairement décider des étapes suivantes du cycle de vie des bugs qu'ils ont soumis.

Les délais dans lesquels la prise en charge des bugs est assurée n'est en général pas garantie une fois les bugs saisis.

Les personnes extérieures aux groupes d'utilisateurs autorisés des bugtrackers ont ainsi la nécessité de « resynchroniser » leurs processus propres avec les processus en cours dans les projets qu'ils utilisent.

Aujourd'hui un tel suivi est essentiellement géré de façon manuelle, et repose sur des fonctions de notification le plus souvent par email. Les bugtrackers disposent de la capacité d'envoyer des courriels relatifs à l'évolution d'un bug à la liste de personnes qui se sont abonnées au suivi de ce bug. Les pratiques actuelles font qu'une personne qui soumet un bug, demande à être notifiée si elle désire suivre l'évolution de son rapport.

Des applications de *recherche*, *d'annotation* et de *notification* de changement de certaines propriétés (notamment le statut représentant les différents états du cycle de vie des bugs) seraient ainsi nécessaires, afin de permettre aux soumissionnaires de surveiller des listes de bugs qui les intéressent, selon leurs critères personnels (ou par extension pour des groupes d'utilisateurs partageant les mêmes critères : processus de projets).


Une application sera réalisée afin de permettre la **collection et la restitution de listes de bugs à partir de recherches multi-critères sur leurs propriétés**<sup>14</sup>. Cette application gèrera :

- les *propriétés intrinsèques*, telles que définies dans les bugtrackers,
- ainsi que des propriétés définies par l'utilisateur lors d'*annotations sémantiques* de rapports de bugs préalablement identifiés.

Une application d'**annotation sémantique de rapports de bugs externes**<sup>15</sup> sera ainsi développée.

<sup>14</sup> Feature 1 : retrieval of aggregated lists of bugs based on multi-criteria queries

<sup>15</sup> Feature 2 : semantic annotation of external bug reports

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

Une **application de notification** sur la base des **changements constatés dans les résultats de recherche multi-critères**<sup>16</sup> sera réalisée. En complément de l'application de restitution décrite ci-dessus (pull), des alertes (push) pourront être transmises (par mail / flux) par un système de vérification périodique (selon des critères programmés à l'avance). L'utilisateur pourra ainsi être informé automatiquement quand certains critères de recherche renverront des résultats différents de ceux obtenus lors de la vérification précédente.

Ces applications sont génériques, et permettent de répondre à de nombreux cas d'utilisation plus particuliers, sur la base de la recherche de certains critères sémantiques précis liés aux rapports de bugs.

Nous détaillons ci-après différents cas d'utilisation ou de spécialisation de ces applications.

### 7.3.2 Suivi des liens entre bugs identifiés sur les mêmes paquetages logiciels

Les applications présentées ci-dessus peuvent être mises en pratique dans le contexte de la diffusion et de l'utilisation de paquetages logiciels.

Différents groupes peuvent s'intéresser au suivi des rapports de bugs soumis sur un même paquetage, mais dans différents bugtrackers. C'est utile aux utilisateurs des paquetages en question, aux mainteneurs/packageurs des mêmes paquetages pour différentes distributions, aux développeurs amont qui veulent un retour de la part des distributions.

Les recherches peuvent ainsi porter sur :

- l'apparition de nouveaux bugs saisis,
- la disparition de bugs (fermés),
- des patches soumis pour corrections,
- l'identification des doublons,
- l'identification de liens entre bugs dépendants les uns des autres (correction dépendant d'une autre correction),
- l'analyse des traces d'exécution pour, par exemple, rapprocher des bugs et éventuellement détecter des doublons.

Ces différents éléments peuvent s'exprimer par différentes propriétés ou liens (sémantiques) extraits des bugtrackers ou définis par les utilisateurs.

Les entités principalement concernées dans ce type d'utilisation sont les paquetages logiciels et les bugs (cf. « spécification de modèle générique de suivi d'anomalie »)


Nous n'envisageons pas, dans le cadre du projet, de mécanisme d'automatisation ayant un impact sur les bugtrackers pour changer l'état des bugs concernés : le système se contente d'assister les experts concernés (par la facilitation des recherches ou par des alertes).

Les utilisateurs restent experts et souverains dans la modification des bugs dans les différents bugtrackers concernés.

Dans ce contexte, une application dérivée du système de recherche et de notification générique sera

---

<sup>16</sup> Feature 3 : notification of changes

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

réalisée, dans le cadre du projet, afin de **supporter le fonctionnement de l'outil bts-link et pour le rendre plus largement utilisable dans n'importe quel projet d'empaquetage**<sup>17</sup>. Bts-link notifie les mainteneurs des paquetages de la distribution Debian des évolutions qui surviennent sur les bugs des projets amont. Les correspondances avec ces projets amonts ont été identifiées préalablement par ces mêmes mainteneurs. L'objectif dans le cadre du WP3 est de rendre bts-link plus indépendant du bugtracker sous-jacent afin qu'il puisse également être utilisé comme moteur pour fluidifier le suivi de bugs dans Helios mais aussi par d'autres projets en dehors de Debian.

Il existe de nombreuses distributions réalisant l'empaquetage des mêmes logiciels, donc une masse importante de bugs potentiellement dupliqués (en plus de ceux du projet amont), rapportés individuellement dans chaque bugtracker de distribution par sa communauté d'utilisateurs. Ces différents corpus de bug gagneraient à être corréllés entre distributions pour maximiser les chances d'amélioration de l'expertise collaborative (triage, expertise) autour de leur résolution. La capacité d'annotation sémantique par l'ensemble des utilisateurs pour indiquer des liens entre de tels bugs devrait permettre d'envisager le signalement de ces bugs aux mainteneurs (seuls utilisateurs « autorisés » des bugtrackers), et d'élargir ainsi la capacité d'expertise à un plus grand nombre, afin de leur faciliter la résolution des problèmes. Nous ne développerons pas, dans le cadre du WP3, une exploitation de tels liens, réservant de tels développement applicatifs dans le futur (ou via des contributions externes).

### 7.3.3 Miroir de bugtracker / duplications de bugs

Dans le contexte particulier d'un projet, il peut s'avérer nécessaire d'effectuer une distribution d'un même ensemble de rapports de bugs dans plusieurs bugtrackers différents, afin de maintenir des miroirs de bugs, par exemple pour diminuer la dépendance vis-à-vis des liens réseau, ou pour des questions de gestion de projet globale (fork), ou pour certains besoins de confidentialité (bugs de sécurité, etc.).

Il nous apparaît cependant que de tels besoins mettent essentiellement en évidence la carence de certaines fonctions dans les bugtrackers utilisés (bugtrackers distribués, confidentialité de certains attributs, fonctions d'import/export, etc.).

Bien que les technologies développées dans le cadre du projet puissent répondre à ce type de besoin, notamment le format d'échange standard basé sur une ontologie générique et RDF, nous ne nous y consacrerons pas dans le cadre du WP3.

### 7.3.4 Gestion de dépendances entre anomalies / composants logiciels

Le graphe de dépendances entre composants logiciels, du point de vue de l'architecture d'une application, implique un graphe de dépendances de certains rapports de bugs qui portent sur différents composants de cette même application. Ce type d'interaction se matérialise par le fait que :

- le code soit dupliqué; certaines applications embarquent en effet des éléments de code identiques, un rapport de bug concernant cette partie doit pouvoir facilement s'attacher aux

<sup>17</sup> Feature 4 : more generic bts-link tool



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence : LI-15-RAP
Version 1.0 du 16/09/09

autres applications qui utilisent ce code ;

- ou qu'il soit mutualisé par l'utilisation de bibliothèques. Les différentes applications nécessitent parfois une nouvelle construction pour prendre en compte la dépendance statique ou dynamique par rapport aux évolutions de la bibliothèque.

Ce type de dépendances est particulièrement courant quand un bug est signalé pour une application finale, mais qu'il correspond en réalité à un problème présent dans une des bibliothèques utilisées par l'application. Le bug doit donc être résolu par une autre équipe de maintenance. Une stratégie pour la gestion de tels bugs peut consister à réaffecter le bug vers le composant « fautif », ou à saisir un nouveau bug pour ce composant, et à garder un lien de dépendance entre celui saisi originellement sur l'application et le nouveau bug ainsi créé sur la bibliothèque.

Les liens entre bugs doivent aussi être mis en évidence dans le cas de failles de sécurité présentes du fait de la mauvaise utilisation d'une API, et nécessitant un correctif dans toutes les applications utilisatrices de la bibliothèque fautive. Les alertes de sécurité doivent tracer explicitement ce type de lien afin de gérer correctement la gravité des anomalies et de rendre prioritaire la diffusion de versions corrigées de tous les composants impactés.

Le couplage entre les modèles de bugs et les modèles de définition des dépendances applicatives sera utilisé pour construire **une agrégation des listes de bugs identifiés sur l'ensemble des dépendances d'un composant logiciel**<sup>18</sup>, dans le cadre de l'intégration des technologies développées avec le reste de la plate-forme d'intégration du WP4.

### 7.3.5 Gestion des branches de développement


Le suivi des anomalies au cours de la fabrication et de la publication des versions successives d'une même application ou d'un même composant logiciel peut obéir à différentes stratégies de gestion de versions et de publication.

Différents modèles de branches, de versions (stables, de qualification, mineures, etc.) peuvent être utilisés pour l'application d'une méthodologie d'assurance qualité. Il n'existe pas un seul modèle général pour ce faire, ni une seule façon de gérer le couplage entre les anomalies constatées et les versions dans lesquelles elles sont présentes ou corrigées ( et donc les tests de vérification de correction, de non-regression, etc.).

Ce couplage entre versions et anomalies peut être géré différemment dans les bugtrackers, avec, par exemple, l'une des deux stratégies suivantes :

- les rapports de bugs sont saisis une seule fois dans le bugtracker, pour une même anomalie, et des liens entre versions présentant les dysfonctionnements, ou comportant un correctif sont gérés par le bugtracker (ainsi qu'une gestion des suivis des arbres de versions successives), par exemple dans debbugs, pour le cas de la distribution Debian.
- ou, à l'opposé, l'ensemble des bugs toujours présents dans une version donnée est dupliqué lorsqu'est créée la version suivante, donnant lieu à l'existence d'ensembles de bugs distincts pour chaque version. Chaque test effectué permet de clore un bug et un seul. Des bugs

<sup>18</sup> Feature 5 : Component dependencies based bug monitoring

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

correspondant à un correctif similaire, appliqué sur deux branches distinctes pour corriger une même anomalie constatées dans ces deux branches, seront gérés de façon individuelle.

**Le couplage entre bugs et versions sera géré dans le cadre du projet, aux besoins de l'intégration dans le cadre de la plate-forme d'intégration du WP4<sup>19</sup>**, mais sans envisager des fonctionnalités applicatives génériques permettant d'effectuer du suivi entre versions dans tous les cas possibles, pour tout modèle de suivi de version.

Les bugtrackers distribués qui visent à gérer le code et les bugs dans les mêmes référentiels distribués (type bzt ou git), comme bugseverywhere, ou dist-bugs, ne font pas l'objet de notre étude. Ils ne sont pas suffisamment développés actuellement pour avoir fait l'objet d'une évaluation dans le cadre de la définition d'un modèle générique d'anomalie.

---

19 Feature 6 : versions – bugs links monitoring



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence : LI-15-RAP
Version 1.0 du 16/09/09

## 8 Interfaces of a global bug monitoring system

As explained in the previous section, we intend to develop, as part of WP3 a generic-purpose core system that will help to access and monitor bugs that are managed in external bugtrackers, in order to offer a services in the form of Web Services accessible by applications, or directly as a Web interface.


We'll develop also a set of first applications using this system in order to fulfill some particular needs. Some of them will be integrated together in the default Web interface, and some may be provided as other distinct Web applications.

### 8.1 Interfaces of the core generic system

Many of the system's expected features may be better understood with an analogy to RSS feeds: the system would be a collaborative web-based RSS aggregator like Google's reader or FriendFeed.

The following features are identified for such a system (most of them will be identified as relating to *feature 1* "retrieval of aggregated lists of bugs based on multi-criteria queries" described in the previous section) :

- collect facts about bugs managed by remote bugtrackers, to add them to a common bug-related facts/properties database:
  - either using custom retrievers for each tool's interfaces,
  - or using standard interchange formats (like the proposed standard based on RDF and EvoOnt BOM ontology);
- handle the refreshing / purging of these bug facts so that they're reasonably new and thus useful for users of the system
- offer a semantic query interface on the database of bug facts :
  - to local users in Web UI,
  - or to tools through an API, for instance through SPARQL language queries
  - about :
    - facts retrieved from the cache of bugs retrieved from the remote bugtrackers
    - and annotations provided by users,
    - but also on other metadata of the system;
- provide an API for submitting semantic annotations on bugs (by users of the system)
- provide feeds (for instance in the form of RDF feeds containing RSS 1.0 plus other ontologies) corresponding to the queries interface, to be consumed by other applications
- provide AJAX compatible server-side representation of facts from the database (lists and individual resources properties) to be consumed by other client-side Web applications

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

- provide a Web interface for users of the system in order to render contents of the system's database for selected registered users :
  - displaying contents of the database of bugs facts
  - displaying users related informations (“my bugs” like view)
  - displaying informations on bugs related to software packages (“software package” like view)
  - displaying additional community-oriented features (reputation of users, etc.)
- provide a Web interface to create semantic annotations on entities of the database (bugs, packages, etc.) (corresponding to feature 2 “semantic annotation of external bug reports” of the previous section);
- provide a notification system (corresponding to feature 3 “notification of changes” of the previous section) :
  - directly in the web interface or through other communication mechanisms (feeds, email, etc.)
  - concerning linked bugs (status monitoring ala bts-link)
  - concerning certain alerts configured by users as queries, with corresponding thresholds verifications

The Web interfaces related to user's preferences views (“my bugs”) will help provide a dedicated web application acting as a central proxy to the various bugtrackers for many users, helping them retrieve and monitor the bugs they care for, independently of the bugtrackers where they were submitted and are managed. This dedicated web application will be deployable on a specific domain on the Web, as part of WP3.

The APIs and/or the AJAX compatible interfaces will help provide the necessary data to be rendered in the bugs facets of the documentation portal of the Helios platform implemented as part of WP 2 (2.2 sub-part).


The annotations may be provided by the WP2's *doc4* components of the Helios platform, or by client tools in relation to the Nepomuk project.

The RDF feeds of the query APIs will be available to be used by other applications, either Web applications or desktop clients.

## 8.2 First complementary applications

A complementary notification application will be developed, which will use the interfaces of the core system to provide a *bug status* monitoring system, for the specific case of interlinked bugs that live along the release/packaging streams (Cf. Feature 4 “more generic bts-link tool” in the previous section).

Two other particular applications will be developed in order to integrate with the rest of the Helios

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

integration platform (Cf. WP4), to provide :

- dependent bug monitoring features based on component dependencies (see feature 5 in previous section),
- bug tracking along version trees for the same components (see feature 6 in the previous section)

Other applications may be developed by third parties, exploiting the interfaces of the core system, in order to fit their particular needs.



## 9 General architecture

The following diagram (Figure 9) illustrates the architecture of the system.

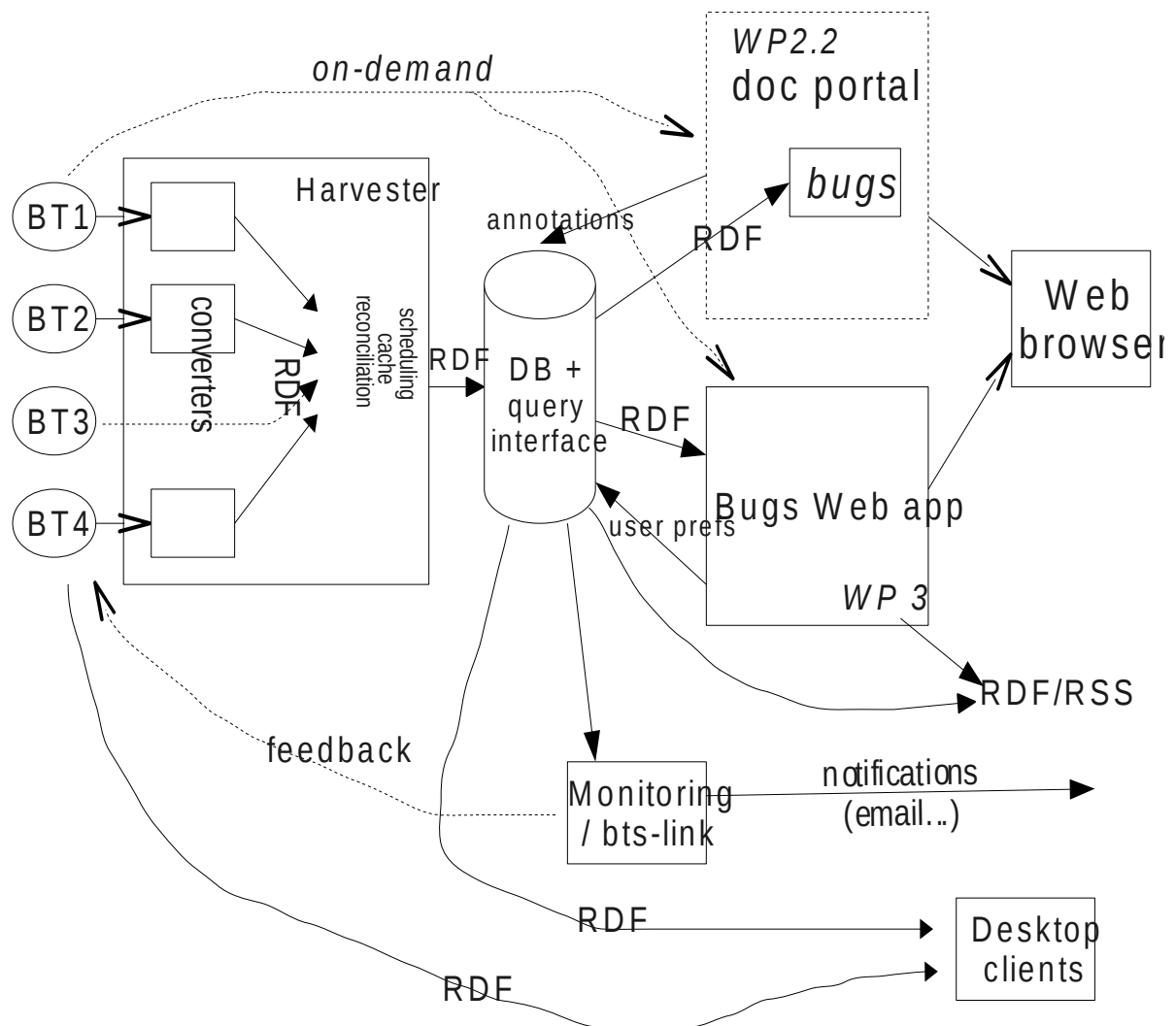


Figure 9: General architecture diagram

The system contains the following components :

- a harvester system that can :
  - launch harvesters and converters that can periodically collect data on the bugtracker interfaces, and save it to a standard format (RDF)




Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence : LI-15-RAP
Version 1.0 du 16/09/09

- schedule the harvesting by the harvesters depending on data freshness in the DB's cache
- reconciles parts of data between the different bugtrackers that refer to same entities (packages, users, bugs, etc.)
- a facts database, storing semantic data for :
  - bug facts
  - people facts
  - software package facts
  - annotations on these
  - administrative meta-data for the system (dates of collection, source of data, etc.)
- a query interface for the database :
  - supporting the SPARQL language for semantic queries,
  - and outputting RDF results to be consumed by :
    - other components of the system containing
    - other applications (for public data)
- a monitoring system that can :
  - perform monitoring of facts in the database in order to notify interested parties of some detected situations,
  - prepare feedback (for instance based on analysis of data or annotations) to be provided back to bugtrackers
- A Web application available for users on the Web (standalone) which can :
  - allow users to access the whole database to retrieve bugs they are interested in
  - perform additional custom queries to bugtrackers
  - store user preferences and User Generated Content (annotations, co-marking, etc.) to the database
  - provide RDF/RSS feeds corresponding to their queries/preferences for export to user's applications
- Web components that can :
  - render data retrieved on-demand from bugtrackers or from the database cache
  - be embedded into the documentation portal for bugs visualisation
  - provide annotations back to the facts database

#### Additional considerations :

- Some of the components developed for the system may also be reused in desktop systems. For instance, the storage, caching, query and annotation mechanisms may be also useful on

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

client desktops for offline processing and advanced bug management. Such interfaces may benefit from compatibility with the Nepomuk technology for desktop integration, or with Mylyn for integration with the Eclipse IDE.

- Some Web interfaces could be developed as AJAX technology, allowing the embedding of bug representation widgets in other applications (Web or desktop/client).



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence : LI-15-RAP
Version 1.0 du 16/09/09

## 10 Interfaces with bugtrackers

Some bugtrackers offer APIs for interfacing with them, like Mantis' SOAP interface.

Some bugtrackers offer native export in structured formats like Bugzilla's XML (and RDF, although not really satisfying as of bugzilla 3.4) feeds, to retrieve lists of bugs or individual bugs descriptions.

They may not always offer full access / export of all the bugs properties, and no unified format exists for doing so.

Some bugtrackers need to be accessed with “web-scraping” harvesters that parse the content of the Web interfaces of the bugtrackers (as is done in some Mylyn connectors, or in bts-link).

There are lots of libraries or tools interfacing with different bugtrackers, which may be reused to implement the bug harvesters :


Tool / library	Program ming language	Supported bugtrackers					
		Bugzilla	Mantis	Debbugs	Launchpad	Fusionforge	Trac
bts-link	Python						

XXX TODO : to be completed XXX

The individual bugtracker converters may all output complete or partial RDF resource descriptions for the retrieved bugs, which could then all be aggregated into the same database.

Note that not all attributes may need to be converted for the system to still be able to operate. Additional attributes may be retrieved on-demand when necessary, as long as key core attributes about bugs, people and software package have been collected.

Still, some complete converters may be necessary, for instance to offer additional migration / replication services, that may also be needed for integration with the rest of the Helios platform.

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09


## 11 Integration with WP 2's documentation portal

Bugs are just one of the many entities that participate to the documentation of software components/packages. Some facts known by the generic bug monitoring system (mainly links between bugs and package versions) will be particularly interesting to WP2's doc4 heterogenous documentation portal.

Any view / query result about bug facts made on the bugs facts database (through the interfaces presented in 8.1 ) may be integrated/replicated into the WP2 portal in this respect.

Query interfaces and annotation interfaces will be developed jointly between WP2 and WP3.

The WP2 portal and some of the WP3 Web Uis will be developed using the same Xwiki-based technology.

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

## 12 Integration with the rest of the Helios platform


The design of the system proposed above is targetting a general case where such a system may be deployed as a standalone application (or, for some components as part of the WP2's documentation portal).

It may also be deployed as a particular instance running in the frame of the Helios platform, in particular in order to monitor bugs of the platform's bugtracker (Mantis), for users of the platform (shared user directory), and concerning software components of projects instanciated in Helios (to track component dependency and version trees in the bugs database).

As a consequence, the Mantis harvester components may be covering the most complete set of attributes of Mantis bugs.

The sharing of the same semantic database with other tools of the platform may help integration (see WP4).

The sharing of the system's metadata model may also help integration (see WP4).

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

## Annex : Helios\_BT reference documentation

The contents of this section is a copy of the automatically-generated documentation of the ontology, as output with Protege's OWLDoc tool, and later processed for inclusion in this documentation, with a tool developped for the needs of Helios.

### 12.1.1 Ontology: helios\_bt

**12.1.1.1** [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl)

#### 12.1.1.2 References

- [Classes](#) (54)
- [Object Properties](#) (12)
- [Data Properties](#) (1)

#### 12.1.1.3 Annotations

- <http://purl.org/dc/elements/1.1/rights>: "This work is Copyrighted (c) 2009 Olivier Berger, Valentin Vlasceanu and Institut TELECOM. It is released under the terms of the Creative Commons Attribution 2.0 France License (<http://creativecommons.org/licenses/by/2.0/fr/deed.en>). It is based on EvoOnt by the Dynamic and Distributed Information Systems Group, University of Zurich, which is licensed under a Creative Commons Attribution 2.5 Switzerland License (see <http://www.ifi.uzh.ch/ddis/evo>) ." (en)
- comment: "Ontology for bugtracker bugs modeling, intended to complement EvoOnt BOM. The version of bom at <http://baetle.googlecode.com/svn/evoont/trunk/bom/bom.owl> will be the reference considered here. This ontology results from work undertaken in the frame of the HELIOS project (<http://www.helios-platform.org>). " (en)
- versionInfo: "0.1"
- <http://purl.org/dc/elements/1.1/creator>: "Olivier Berger, Valentin Vlasceanu"

#### 12.1.1.4 Imports

- imports: <http://www.w3.org/2005/01/wf/flow>
- imports: <http://baetle.googlecode.com/svn/evoont/trunk/bom/bom.owl>

#### 12.1.1.5 Class Usage

[BugTracker](#) [BugzillaDefect](#) [BugzillaEnhancement](#) [BugzillaPriorityP1](#) [BugzillaPriorityP2](#) [BugzillaPriorityP3](#) [BugzillaPriorityP4](#) [BugzillaPriorityP5](#) [BugzillaResolutionDUPLICATE](#)



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie

Référence : LI-15-RAP

Version 1.0 du 16/09/09

[BugzillaResolutionFIXED](#) [BugzillaResolutionINVALID](#) [BugzillaResolutionMOVED](#)  
[BugzillaResolutionWONTFIX](#) [BugzillaResolutionWORKSFORME](#) [BugzillaSeverityblocker](#)  
[BugzillaSeveritycritical](#) [BugzillaSeverityenhancement](#) [BugzillaSeveritymajor](#)  
[BugzillaSeverityminor](#) [BugzillaSeveritynormal](#) [BugzillaSeveritytrivial](#) [BugzillaStateASSIGNED](#)  
[BugzillaStateCLOSED](#) [BugzillaStateNEW](#) [BugzillaStateREOPENED](#) [BugzillaStateRESOLVED](#)  
[BugzillaStateUNCONFIRMED](#) [BugzillaStateVERIFIED](#) [Component](#) [DebbugsDefect](#)  
[DebbugsEnhancement](#) [DebbugsSeveritycritical](#) [DebbugsSeveritygrave](#) [DebbugsSeverityimportant](#)  
[DebbugsSeverityminor](#) [DebbugsSeveritynormal](#) [DebbugsSeverityserious](#) [DebbugsSeveritywishlist](#)  
[DebbugsStatedone](#) [DebbugsStatefixed](#) [DebbugsStateforwarded](#) [DebbugsStatepending](#)  
[DebbugsStatepending-fixed](#) [Issue](#) [NonTerminalState](#) [Priority](#) [Resolution](#) [Severity](#) [TerminalState](#)

---

### 12.1.1.6 *helios\_bt: classes (54)*

- [Thing](#)
- [BugTracker](#)
- [BugzillaDefect](#)
- [BugzillaEnhancement](#)
- [BugzillaPriorityP1](#)
- [BugzillaPriorityP2](#)
- [BugzillaPriorityP3](#)
- [BugzillaPriorityP4](#)
- [BugzillaPriorityP5](#)
- [BugzillaResolutionDUPLICATE](#)
- [BugzillaResolutionFIXED](#)
- [BugzillaResolutionINVALID](#)
- [BugzillaResolutionMOVED](#)
- [BugzillaResolutionWONTFIX](#)
- [BugzillaResolutionWORKSFORME](#)
- [BugzillaSeverityblocker](#)
- [BugzillaSeveritycritical](#)
- [BugzillaSeverityenhancement](#)
- [BugzillaSeveritymajor](#)
- [BugzillaSeverityminor](#)
- [BugzillaSeveritynormal](#)
- [BugzillaSeveritytrivial](#)
- [BugzillaStateASSIGNED](#)
- [BugzillaStateCLOSED](#)
- [BugzillaStateNEW](#)
- [BugzillaStateREOPENED](#)
- [BugzillaStateRESOLVED](#)
- [BugzillaStateUNCONFIRMED](#)
- [BugzillaStateVERIFIED](#)
- [Component](#)
- [DebbugsDefect](#)
- [DebbugsEnhancement](#)





Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie

Référence : LI-15-RAP

Version 1.0 du 16/09/09

- [DebbugsSeveritycritical](#)
  - [DebbugsSeveritygrave](#)
  - [DebbugsSeverityimportant](#)
  - [DebbugsSeverityminor](#)
  - [DebbugsSeveritynormal](#)
  - [DebbugsSeverityserious](#)
  - [DebbugsSeveritywishlist](#)
  - [DebbugsStatedone](#)
  - [DebbugsStatefixed](#)
  - [DebbugsStateforwarded](#)
  - [DebbugsStatepending](#)
  - [DebbugsStatepending-fixed](#)
  - [Issue](#)
  - [NonTerminalState](#)
  - [Priority](#)
  - [Product](#)
  - [Project](#)
  - [Resolution](#)
  - [Severity](#)
  - [State](#)
  - [TerminalState](#)
  - [Version](#)
- 

### **12.1.1.7 Class: Thing**

#### **12.1.1.7.1 <http://www.w3.org/2002/07/owl#Thing>**

- Thing
  - Action +
  - Activity
  - Agent +
  - Attachment
  - [BugTracker](#)
  - Class +
  - Class
  - Comment
  - Community
  - [Component](#)
  - ComputerSystem
  - Container +
  - DateTime
  - DeprecatedProperty
  - Document +
  - Image



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie

Référence : LI-15-RAP

Version 1.0 du 16/09/09

- Item +
- jabberID
- List
- Literal
- mbox\_sha1sum
- Milestone
- OnlineAccount +
- Person +
- [Priority](#) +
- [Product](#)
- [Project](#) +
- [Project](#)
- [Project](#) +
- Repository +
- [Resolution](#) +
- Resource +
- Role
- [Severity](#) +
- SocialEntity
- Space +
- SpatialThing +
- Task +
- Usergroup
- [Version](#)
- [Version](#)
- [Version](#)
- Work

---

### **12.1.1.8 Class: BugTracker**

**12.1.1.8.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#BugTracker](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#BugTracker)**

- [Thing](#)
- BugTracker

#### **Annotations**

- comment: "helios\_bt ontology class, designed to better describe a bugtracker"

### **12.1.1.8.2 Superclasses**

- [Thing](#)



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence : LI-15-RAP
Version 1.0 du 16/09/09

### 12.1.1.8.3 Usage

- [bugtrackerVersion](#) domain BugTracker
  - [bugTrackerURL](#) domain BugTracker
  - [inBugtracker](#) range BugTracker
  - [bugtracker](#) range BugTracker
- 

### 12.1.1.9 Class: BugzillaDefect

#### 12.1.1.9.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#BugzillaDefect](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#BugzillaDefect)

- [Thing](#)
  - [Issue](#)
    - BugzillaDefect

#### Annotations

- comment: "Complements the bom:Issue class by adding the notion of defect for Bugzilla bugtracker. It has a restriction which only allows it to take values if the severity of the issue is: blocker, critical, major, minor, normal or trivial. For the other values has been defined another class: BugzillaEnhancement."

#### 12.1.1.9.2 Superclasses

- [Issue](#)
- [hasSeverity](#) only ( [BugzillaSeverityblocker](#) or [BugzillaSeveritycritical](#) or [BugzillaSeveritymajor](#) or [BugzillaSeverityminor](#) or [BugzillaSeveritynormal](#) or [BugzillaSeveritytrivial](#) )

#### 12.1.1.9.3 Disjoints

[BugzillaEnhancement](#)

---

### 12.1.1.10 Class: BugzillaEnhancement

#### 12.1.1.10.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#BugzillaEnhancement](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#BugzillaEnhancement)

- [Thing](#)
  - [Issue](#)
    - BugzillaEnhancement



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie

Référence : LI-15-RAP

Version 1.0 du 16/09/09

## Annotations

- comment: "Complements the bom:Issue class by adding the notion of enhancement for Bugzilla bugtracker. It has a restriction which only allows it to take values if the severity of the issue is: enhancement. For the other values has been defined another class: BugzillaDefect."

### 12.1.1.10.2 Superclasses

- [Issue](#)
- [hasSeverity](#) only [BugzillaSeverityenhancement](#)

### 12.1.1.10.3 Disjoints

[BugzillaDefect](#)

---

### 12.1.1.11 Class: BugzillaPriorityP1

12.1.1.11.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#BugzillaPriorityP1](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#BugzillaPriorityP1)

- [Thing](#)
  - [Priority](#)
    - BugzillaPriorityP1

## Annotations

- comment: "Complements the Priority class by adding Bugzilla specific subclass." (en)

### 12.1.1.11.2 Superclasses

- [Priority](#)

### 12.1.1.11.3 Disjoints

[BugzillaPriorityP4](#), [BugzillaPriorityP5](#), [BugzillaPriorityP3](#), [BugzillaPriorityP2](#)

---

### 12.1.1.12 Class: BugzillaPriorityP2

12.1.1.12.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#BugzillaPriorityP2](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#BugzillaPriorityP2)

- [Thing](#)
  - [Priority](#)



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence : LI-15-RAP
Version 1.0 du 16/09/09

- BugzillaPriorityP2

### **Annotations**

- comment: "Complements the Priority class by adding Bugzilla specific subclass."

#### **12.1.1.12.2 Superclasses**

- [Priority](#)

#### **12.1.1.12.3 Disjoints**

[BugzillaPriorityP1](#) , [BugzillaPriorityP4](#) , [BugzillaPriorityP5](#) , [BugzillaPriorityP3](#)

---

#### **12.1.1.13 Class: BugzillaPriorityP3**

**12.1.1.13.1** [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#BugzillaPriorityP3](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#BugzillaPriorityP3)

- [Thing](#)
  - [Priority](#)
    - BugzillaPriorityP3

### **Annotations**

- comment: "Complements the Priority class by adding Bugzilla specific subclass."

#### **12.1.1.13.2 Superclasses**

- [Priority](#)

#### **12.1.1.13.3 Disjoints**

[BugzillaPriorityP1](#) , [BugzillaPriorityP4](#) , [BugzillaPriorityP5](#) , [BugzillaPriorityP2](#)

---

#### **12.1.1.14 Class: BugzillaPriorityP4**

**12.1.1.14.1** [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#BugzillaPriorityP4](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#BugzillaPriorityP4)

- [Thing](#)
  - [Priority](#)
    - BugzillaPriorityP4



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence : LI-15-RAP
Version 1.0 du 16/09/09

### Annotations

- comment: "Complements the Priority class by adding Bugzilla specific subclass."

#### 12.1.1.14.2 Superclasses

- [Priority](#)

#### 12.1.1.14.3 Disjoints

[BugzillaPriorityP1](#), [BugzillaPriorityP5](#), [BugzillaPriorityP3](#), [BugzillaPriorityP2](#)

---

#### 12.1.1.15 Class: BugzillaPriorityP5

12.1.1.15.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#BugzillaPriorityP5](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#BugzillaPriorityP5)

- [Thing](#)
  - [Priority](#)
    - BugzillaPriorityP5

### Annotations

- comment: "Complements the Priority class by adding Bugzilla specific subclass."

#### 12.1.1.15.2 Superclasses

- [Priority](#)

#### 12.1.1.15.3 Disjoints

[BugzillaPriorityP1](#), [BugzillaPriorityP4](#), [BugzillaPriorityP3](#), [BugzillaPriorityP2](#)

---

#### 12.1.1.16 Class: BugzillaResolutionDUPLICATE

12.1.1.16.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#BugzillaResolutionDUPLICATE](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#BugzillaResolutionDUPLICATE)

- [Thing](#)
  - [Resolution](#)
    - BugzillaResolutionDUPLICATE

### Annotations



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence : LI-15-RAP
Version 1.0 du 16/09/09

- comment: "Complements the Resolution class by adding Bugzilla specific subclass."

#### **12.1.1.16.2 Superclasses**

- [Resolution](#)

#### **12.1.1.16.3 Disjoints**

[BugzillaResolutionWONTFIX](#), [BugzillaResolutionINVALID](#), [BugzillaResolutionFIXED](#),  
[BugzillaResolutionWORKSFORME](#), [BugzillaResolutionMOVED](#)

---

#### **12.1.1.17 Class: BugzillaResolutionFIXED**

**12.1.1.17.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#BugzillaResolutionFIXED](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#BugzillaResolutionFIXED)**

- [Thing](#)
  - [Resolution](#)
    - BugzillaResolutionFIXED

#### **Annotations**

- comment: "Complements the Resolution class by adding Bugzilla specific subclass."

#### **12.1.1.17.2 Superclasses**

- [Resolution](#)

#### **12.1.1.17.3 Disjoints**

[BugzillaResolutionWONTFIX](#), [BugzillaResolutionDUPLICATE](#), [BugzillaResolutionINVALID](#),  
[BugzillaResolutionWORKSFORME](#), [BugzillaResolutionMOVED](#)

---

#### **12.1.1.18 Class: BugzillaResolutionINVALID**

**12.1.1.18.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#BugzillaResolutionINVALID](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#BugzillaResolutionINVALID)**

- [Thing](#)
  - [Resolution](#)
    - BugzillaResolutionINVALID

#### **Annotations**



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence : LI-15-RAP
Version 1.0 du 16/09/09

- comment: "Complements the Resolution class by adding Bugzilla specific subclass."

#### **12.1.1.18.2 Superclasses**

- [Resolution](#)

#### **12.1.1.18.3 Disjoints**

[BugzillaResolutionWONTFIX](#), [BugzillaResolutionDUPLICATE](#), [BugzillaResolutionFIXED](#), [BugzillaResolutionWORKSFORME](#), [BugzillaResolutionMOVED](#)

---

#### **12.1.1.19 Class: BugzillaResolutionMOVED**

**12.1.1.19.1** [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#BugzillaResolutionMOVED](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#BugzillaResolutionMOVED)

- [Thing](#)
  - [Resolution](#)
    - BugzillaResolutionMOVED

#### **Annotations**

- comment: "Complements the Resolution class by adding Bugzilla specific subclass."

#### **12.1.1.19.2 Superclasses**

- [Resolution](#)

#### **12.1.1.19.3 Disjoints**

[BugzillaResolutionWONTFIX](#), [BugzillaResolutionDUPLICATE](#), [BugzillaResolutionINVALID](#), [BugzillaResolutionFIXED](#), [BugzillaResolutionWORKSFORME](#)

---

#### **12.1.1.20 Class: BugzillaResolutionWONTFIX**

**12.1.1.20.1** [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#BugzillaResolutionWONTFIX](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#BugzillaResolutionWONTFIX)

- [Thing](#)
  - [Resolution](#)
    - BugzillaResolutionWONTFIX

#### **Annotations**





Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie

Référence : LI-15-RAP

Version 1.0 du 16/09/09

- comment: "Complements the Resolution class by adding Bugzilla specific subclass."

#### **12.1.1.20.2 Superclasses**

- [Resolution](#)

#### **12.1.1.20.3 Disjoints**

[BugzillaResolutionDUPLICATE](#), [BugzillaResolutionINVALID](#), [BugzillaResolutionFIXED](#), [BugzillaResolutionWORKSFORME](#), [BugzillaResolutionMOVED](#)

---

#### **12.1.1.21 Class: BugzillaResolutionWORKSFORME**

**12.1.1.21.1** [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#BugzillaResolutionWORKSFORME](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#BugzillaResolutionWORKSFORME)

- [Thing](#)
  - [Resolution](#)
    - BugzillaResolutionWORKSFORME

#### **Annotations**

- comment: "Complements the Resolution class by adding Bugzilla specific subclass."

#### **12.1.1.21.2 Superclasses**

- [Resolution](#)

#### **12.1.1.21.3 Disjoints**

[BugzillaResolutionWONTFIX](#), [BugzillaResolutionDUPLICATE](#), [BugzillaResolutionINVALID](#), [BugzillaResolutionFIXED](#), [BugzillaResolutionMOVED](#)

---

#### **12.1.1.22 Class: BugzillaSeverityblocker**

**12.1.1.22.1** [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#BugzillaSeverityblocker](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#BugzillaSeverityblocker)

- [Thing](#)
  - [Severity](#)
    - BugzillaSeverityblocker



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie

Référence : LI-15-RAP

Version 1.0 du 16/09/09

## Annotations

- comment: "Complements the Severity class by adding Bugzilla specific subclass."

### 12.1.1.22.2 Superclasses

- [Severity](#)

### 12.1.1.22.3 Usage

- [BugzillaDefect](#) -> [hasSeverity](#) only ( [BugzillaSeverityblocker](#) or [BugzillaSeveritycritical](#) or [BugzillaSeveritymajor](#) or [BugzillaSeverityminor](#) or [BugzillaSeveritynormal](#) or [BugzillaSeveritytrivial](#) )

---

### 12.1.1.23 Class: *BugzillaSeveritycritical*

**12.1.1.23.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#BugzillaSeveritycritical](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#BugzillaSeveritycritical)**

- [Thing](#)
  - [Severity](#)
    - [BugzillaSeveritycritical](#)

## Annotations

- comment: "Complements the Severity class by adding Bugzilla specific subclass."

### 12.1.1.23.2 Superclasses

- [Severity](#)

### 12.1.1.23.3 Usage

- [BugzillaDefect](#) -> [hasSeverity](#) only ( [BugzillaSeverityblocker](#) or [BugzillaSeveritycritical](#) or [BugzillaSeveritymajor](#) or [BugzillaSeverityminor](#) or [BugzillaSeveritynormal](#) or [BugzillaSeveritytrivial](#) )

---

### 12.1.1.24 Class: *BugzillaSeverityenhancement*

**12.1.1.24.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#BugzillaSeverityenhancement](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#BugzillaSeverityenhancement)**

- [Thing](#)



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie

Référence : LI-15-RAP

Version 1.0 du 16/09/09

- [Severity](#)
  - BugzillaSeverityenhancement

### **Annotations**

- comment: "Complements the Severity class by adding Bugzilla specific subclass."

#### **12.1.1.24.2 Superclasses**

- [Severity](#)

#### **12.1.1.24.3 Usage**

- [BugzillaEnhancement](#) -> [hasSeverity](#) only BugzillaSeverityenhancement
- 

#### **12.1.1.25 Class: BugzillaSeveritymajor**

**12.1.1.25.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#BugzillaSeveritymajor](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#BugzillaSeveritymajor)**

- [Thing](#)
  - [Severity](#)
    - BugzillaSeveritymajor

### **Annotations**

- comment: "Complements the Severity class by adding Bugzilla specific subclass."

#### **12.1.1.25.2 Superclasses**

- [Severity](#)

#### **12.1.1.25.3 Usage**

- [BugzillaDefect](#) -> [hasSeverity](#) only ( [BugzillaSeverityblocker](#) or [BugzillaSeveritycritical](#) or [BugzillaSeveritymajor](#) or [BugzillaSeverityminor](#) or [BugzillaSeveritynormal](#) or [BugzillaSeveritytrivial](#) )
- 

#### **12.1.1.26 Class: BugzillaSeverityminor**

**12.1.1.26.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#BugzillaSeverityminor](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#BugzillaSeverityminor)**

- [Thing](#)



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie

Référence : LI-15-RAP

Version 1.0 du 16/09/09

- [Severity](#)
  - BugzillaSeverityminor

### **Annotations**

- comment: "Complements the Severity class by adding Bugzilla specific subclass."

#### **12.1.1.26.2 Superclasses**

- [Severity](#)

#### **12.1.1.26.3 Usage**

- [BugzillaDefect](#) -> [hasSeverity](#) only ( [BugzillaSeverityblocker](#) or [BugzillaSeveritycritical](#) or [BugzillaSeveritymajor](#) or [BugzillaSeverityminor](#) or [BugzillaSeveritynormal](#) or [BugzillaSeveritytrivial](#) )
- 

#### **12.1.1.27 Class: BugzillaSeveritynormal**

##### **12.1.1.27.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#BugzillaSeveritynormal](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#BugzillaSeveritynormal)**

- [Thing](#)
  - [Severity](#)
    - BugzillaSeveritynormal

### **Annotations**

- comment: "Complements the Severity class by adding Bugzilla specific subclass."

#### **12.1.1.27.2 Superclasses**

- [Severity](#)

#### **12.1.1.27.3 Usage**

- [BugzillaDefect](#) -> [hasSeverity](#) only ( [BugzillaSeverityblocker](#) or [BugzillaSeveritycritical](#) or [BugzillaSeveritymajor](#) or [BugzillaSeverityminor](#) or [BugzillaSeveritynormal](#) or [BugzillaSeveritytrivial](#) )
-



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence : LI-15-RAP
Version 1.0 du 16/09/09

### **12.1.1.28 Class: *BugzillaSeveritytrivial***

**12.1.1.28.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#BugzillaSeveritytrivial](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#BugzillaSeveritytrivial)**

- [Thing](#)
  - [Severity](#)
    - *BugzillaSeveritytrivial*

#### **Annotations**

- comment: "Complements the Severity class by adding Bugzilla specific subclass."

### **12.1.1.28.2 Superclasses**

- [Severity](#)

### **12.1.1.28.3 Usage**

- [BugzillaDefect](#) -> [hasSeverity](#) only ( [BugzillaSeverityblocker](#) or [BugzillaSeveritycritical](#) or [BugzillaSeveritymajor](#) or [BugzillaSeverityminor](#) or [BugzillaSeveritynormal](#) or *BugzillaSeveritytrivial* )

### **12.1.1.29 Class: *BugzillaStateASSIGNED***

**12.1.1.29.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#BugzillaStateASSIGNED](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#BugzillaStateASSIGNED)**

- [Thing](#)
  - [State](#)
    - [NonTerminalState](#)
      - *BugzillaStateASSIGNED*

#### **Annotations**

- comment: "Complements the NonTerminalState class by adding Bugzilla specific subclass."

### **12.1.1.29.2 Superclasses**

- [NonTerminalState](#)

### **12.1.1.29.3 Disjoints**

[BugzillaStateCLOSED](#), [BugzillaStateUNCONFIRMED](#), [BugzillaStateVERIFIED](#), [BugzillaStateRESOLVED](#), [BugzillaStateNEW](#), [BugzillaStateREOPENED](#)



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence : LI-15-RAP
Version 1.0 du 16/09/09

### **12.1.1.30 Class: BugzillaStateCLOSED**

**12.1.1.30.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#BugzillaStateCLOSED](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#BugzillaStateCLOSED)**

- [Thing](#)
  - [TerminalState](#)
    - BugzillaStateCLOSED

#### **Annotations**

- comment: "Complements the TerminalState class by adding Bugzilla specific subclass."

### **12.1.1.30.2 Superclasses**

- [TerminalState](#)

### **12.1.1.30.3 Disjoints**

[BugzillaStateUNCONFIRMED](#), [BugzillaStateVERIFIED](#), [BugzillaStateRESOLVED](#), [BugzillaStateNEW](#), [BugzillaStateASSIGNED](#), [BugzillaStateREOPENED](#)

---

### **12.1.1.31 Class: BugzillaStateNEW**

**12.1.1.31.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#BugzillaStateNEW](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#BugzillaStateNEW)**

- [Thing](#)
  - [State](#)
    - [NonTerminalState](#)
      - BugzillaStateNEW

#### **Annotations**

- comment: "Complements the NonTerminalState class by adding Bugzilla specific subclass."

### **12.1.1.31.2 Superclasses**

- [NonTerminalState](#)

### **12.1.1.31.3 Disjoints**

[BugzillaStateCLOSED](#), [BugzillaStateUNCONFIRMED](#), [BugzillaStateVERIFIED](#), [BugzillaStateRESOLVED](#), [BugzillaStateASSIGNED](#), [BugzillaStateREOPENED](#)

---



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence : LI-15-RAP
Version 1.0 du 16/09/09

### **12.1.1.32 Class: *BugzillaStateREOPENED***

**12.1.1.32.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#BugzillaStateREOPENED](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#BugzillaStateREOPENED)**

- [Thing](#)
  - [State](#)
    - [NonTerminalState](#)
      - *BugzillaStateREOPENED*

#### **Annotations**

- comment: "Complements the NonTerminalState class by adding Bugzilla specific subclass."

### **12.1.1.32.2 Superclasses**

- [NonTerminalState](#)

### **12.1.1.32.3 Disjoints**

[BugzillaStateCLOSED](#), [BugzillaStateUNCONFIRMED](#), [BugzillaStateVERIFIED](#),  
[BugzillaStateRESOLVED](#), [BugzillaStateNEW](#), [BugzillaStateASSIGNED](#)

---

### **12.1.1.33 Class: *BugzillaStateRESOLVED***

**12.1.1.33.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#BugzillaStateRESOLVED](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#BugzillaStateRESOLVED)**

- [Thing](#)
  - [State](#)
    - [NonTerminalState](#)
      - *BugzillaStateRESOLVED*

#### **Annotations**

- comment: "Complements the NonTerminalState class by adding Bugzilla specific subclass."

### **12.1.1.33.2 Superclasses**

- [NonTerminalState](#)

### **12.1.1.33.3 Disjoints**

[BugzillaStateCLOSED](#), [BugzillaStateUNCONFIRMED](#), [BugzillaStateVERIFIED](#),  
[BugzillaStateNEW](#), [BugzillaStateASSIGNED](#), [BugzillaStateREOPENED](#)

---



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence : LI-15-RAP
Version 1.0 du 16/09/09

#### **12.1.1.34 Class: BugzillaStateUNCONFIRMED**

**12.1.1.34.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#BugzillaStateUNCONFIRMED](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#BugzillaStateUNCONFIRMED)**

- [Thing](#)
  - [State](#)
    - [NonTerminalState](#)
      - BugzillaStateUNCONFIRMED

#### **Annotations**

- comment: "Complements the NonTerminalState class by adding Bugzilla specific subclass."

#### **12.1.1.34.2 Superclasses**

- [NonTerminalState](#)

#### **12.1.1.34.3 Disjoints**

[BugzillaStateCLOSED](#), [BugzillaStateVERIFIED](#), [BugzillaStateRESOLVED](#), [BugzillaStateNEW](#), [BugzillaStateASSIGNED](#), [BugzillaStateREOPENED](#)

---

#### **12.1.1.35 Class: BugzillaStateVERIFIED**

**12.1.1.35.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#BugzillaStateVERIFIED](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#BugzillaStateVERIFIED)**

- [Thing](#)
  - [State](#)
    - [NonTerminalState](#)
      - BugzillaStateVERIFIED

#### **Annotations**

- comment: "Complements the NonTerminalState class by adding Bugzilla specific subclass."

#### **12.1.1.35.2 Superclasses**

- [NonTerminalState](#)

#### **12.1.1.35.3 Disjoints**

[BugzillaStateCLOSED](#), [BugzillaStateUNCONFIRMED](#), [BugzillaStateRESOLVED](#), [BugzillaStateNEW](#), [BugzillaStateASSIGNED](#), [BugzillaStateREOPENED](#)

---





Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie

Référence : LI-15-RAP

Version 1.0 du 16/09/09

### **12.1.1.36 Class: Component**

#### **12.1.1.36.1 <http://www.ifi.uzh.ch/ddis/evoont/2008/11/bom#Component>**

- [Thing](#)
  - Component

#### **Usage**

- [hasIssue](#) domain Component
  - [isIssueOf](#) range Component
- 

### **12.1.1.37 Class: DebbugsDefect**

#### **12.1.1.37.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#DebbugsDefect](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#DebbugsDefect)**

- [Thing](#)
  - [Issue](#)
    - DebbugsDefect

#### **Annotations**

- comment: "Complements the bom:Issue class by adding the notion of defect for debbugs bugtracker. It has a restriction which only allows it to take values if the severity of the issue is: critical, grave, important, minor, normal, serious. For the other values has been defined another class: DebbugsEnhancement."

#### **12.1.1.37.2 Superclasses**

- [Issue](#)
- [hasSeverity](#) only ( [DebbugsSeveritycritical](#) or [DebbugsSeveritygrave](#) or [DebbugsSeverityimportant](#) or [DebbugsSeverityminor](#) or [DebbugsSeveritynormal](#) or [DebbugsSeverityserious](#) )

#### **12.1.1.37.3 Disjoints**

[DebbugsEnhancement](#)

---



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence : LI-15-RAP
Version 1.0 du 16/09/09

### **12.1.1.38 Class: *DebbugsEnhancement***

**12.1.1.38.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#DebbugsEnhancement](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#DebbugsEnhancement)**

- [Thing](#)
  - [Issue](#)
    - *DebbugsEnhancement*

#### **Annotations**

- comment: "Enhancements for wishlists Complements the bom:Issue class by adding the notion of enhancement for debbugs bugtracker. It has a restriction which only allows it to take values if the severity of the issue is: wishlist. For the other values has been defined another class: *DebbugsDefect*."

### **12.1.1.38.2 Superclasses**

- [Issue](#)
- [hasSeverity](#) only [DebbugsSeveritywishlist](#)

### **12.1.1.38.3 Disjoints**

[DebbugsDefect](#)

---

### **12.1.1.39 Class: *DebbugsSeveritycritical***

**12.1.1.39.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#DebbugsSeveritycritical](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#DebbugsSeveritycritical)**

- [Thing](#)
  - [Severity](#)
    - *DebbugsSeveritycritical*

#### **Annotations**

- comment: "Complements the Severity class by adding debbugs specific subclass."

### **12.1.1.39.2 Superclasses**

- [Severity](#)

### **12.1.1.39.3 Usage**

- [DebbugsDefect](#) -> [hasSeverity](#) only ( *DebbugsSeveritycritical* or [DebbugsSeveritygrave](#) or [DebbugsSeverityimportant](#) or [DebbugsSeverityminor](#) or [DebbugsSeveritynormal](#) or



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie

Référence : LI-15-RAP

Version 1.0 du 16/09/09

[DebbugsSeverityserious](#))

---

#### **12.1.1.40 Class: DebbugsSeveritygrave**

**12.1.1.40.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#DebbugsSeveritygrave](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#DebbugsSeveritygrave)**

- [Thing](#)
  - [Severity](#)
    - DebbugsSeveritygrave

#### **Annotations**

- comment: "Complements the Severity class by adding debbugs specific subclass."

#### **12.1.1.40.2 Superclasses**

- [Severity](#)

#### **12.1.1.40.3 Usage**

- [DebbugsDefect](#) -> [hasSeverity](#) only ( [DebbugsSeveritycritical](#) or [DebbugsSeveritygrave](#) or [DebbugsSeverityimportant](#) or [DebbugsSeverityminor](#) or [DebbugsSeveritynormal](#) or [DebbugsSeverityserious](#) )
- 

#### **12.1.1.41 Class: DebbugsSeverityimportant**

**12.1.1.41.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#DebbugsSeverityimportant](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#DebbugsSeverityimportant)**

- [Thing](#)
  - [Severity](#)
    - DebbugsSeverityimportant

#### **Annotations**

- comment: "Complements the Severity class by adding debbugs specific subclass."

#### **12.1.1.41.2 Superclasses**

- [Severity](#)

#### **12.1.1.41.3 Usage**



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence : LI-15-RAP
Version 1.0 du 16/09/09

- [DebbugsDefect](#) -> [hasSeverity](#) only ( [DebbugsSeveritycritical](#) or [DebbugsSeveritygrave](#) or [DebbugsSeverityimportant](#) or [DebbugsSeverityminor](#) or [DebbugsSeveritynormal](#) or [DebbugsSeverityserious](#) )
- 

#### **12.1.1.42 Class: DebbugsSeverityminor**

**12.1.1.42.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#DebbugsSeverityminor](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#DebbugsSeverityminor)**

- [Thing](#)
  - [Severity](#)
    - DebbugsSeverityminor

#### **Annotations**

- comment: "Complements the Severity class by adding debbugs specific subclass."

#### **12.1.1.42.2 Superclasses**

- [Severity](#)

#### **12.1.1.42.3 Usage**

- [DebbugsDefect](#) -> [hasSeverity](#) only ( [DebbugsSeveritycritical](#) or [DebbugsSeveritygrave](#) or [DebbugsSeverityimportant](#) or [DebbugsSeverityminor](#) or [DebbugsSeveritynormal](#) or [DebbugsSeverityserious](#) )
- 

#### **12.1.1.43 Class: DebbugsSeveritynormal**

**12.1.1.43.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#DebbugsSeveritynormal](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#DebbugsSeveritynormal)**

- [Thing](#)
  - [Severity](#)
    - DebbugsSeveritynormal

#### **Annotations**

- comment: "Complements the Severity class by adding debbugs specific subclass."

#### **12.1.1.43.2 Superclasses**

- [Severity](#)



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence : LI-15-RAP
Version 1.0 du 16/09/09

### 12.1.1.43.3 Usage

- [DebbugsDefect](#) -> [hasSeverity](#) only ( [DebbugsSeveritycritical](#) or [DebbugsSeveritygrave](#) or [DebbugsSeverityimportant](#) or [DebbugsSeverityminor](#) or [DebbugsSeveritynormal](#) or [DebbugsSeverityserious](#) )
- 

### 12.1.1.44 Class: *DebbugsSeverityserious*

**12.1.1.44.1** [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#DebbugsSeverityserious](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#DebbugsSeverityserious)

- [Thing](#)
  - [Severity](#)
    - *DebbugsSeverityserious*

#### Annotations

- comment: "Complements the Severity class by adding debbugs specific subclass."

### 12.1.1.44.2 Superclasses

- [Severity](#)

### 12.1.1.44.3 Usage

- [DebbugsDefect](#) -> [hasSeverity](#) only ( [DebbugsSeveritycritical](#) or [DebbugsSeveritygrave](#) or [DebbugsSeverityimportant](#) or [DebbugsSeverityminor](#) or [DebbugsSeveritynormal](#) or [DebbugsSeverityserious](#) )
- 

### 12.1.1.45 Class: *DebbugsSeveritywishlist*

**12.1.1.45.1** [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#DebbugsSeveritywishlist](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#DebbugsSeveritywishlist)

- [Thing](#)
  - [Severity](#)
    - *DebbugsSeveritywishlist*

#### Annotations

- comment: "Complements the Severity class by adding debbugs specific subclass."

### 12.1.1.45.2 Superclasses



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie

Référence : LI-15-RAP

Version 1.0 du 16/09/09

- [Severity](#)

#### **12.1.1.45.3 Usage**

- [DebbugsEnhancement](#) -> [hasSeverity](#) only DebbugsSeveritywishlist
- 

#### **12.1.1.46 Class: DebbugsStatedone**

**12.1.1.46.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#DebbugsStatedone](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#DebbugsStatedone)**

- [Thing](#)
  - [TerminalState](#)
    - DebbugsStatedone

#### **Annotations**

- comment: "Only state in UDD bugs.status that we'll consider terminal" (en)

#### **12.1.1.46.2 Superclasses**

- [TerminalState](#)
- 

#### **12.1.1.47 Class: DebbugsStatefixed**

**12.1.1.47.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#DebbugsStatefixed](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#DebbugsStatefixed)**

- [Thing](#)
  - [State](#)
    - [NonTerminalState](#)
      - DebbugsStatefixed

#### **Annotations**

- comment: "Complements the NonTerminalState class by adding debbugs specific subclass."

#### **12.1.1.47.2 Superclasses**

- [NonTerminalState](#)
-



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence : LI-15-RAP
Version 1.0 du 16/09/09

#### **12.1.1.48 Class: *DebbugsStateforwarded***

**12.1.1.48.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#DebbugsStateforwarded](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#DebbugsStateforwarded)**

- [Thing](#)
  - [State](#)
    - [NonTerminalState](#)
      - *DebbugsStateforwarded*

#### **Annotations**

- comment: "Complements the NonTerminalState class by adding debbugs specific subclass."

#### **12.1.1.48.2 Superclasses**

- [NonTerminalState](#)
- 

#### **12.1.1.49 Class: *DebbugsStatepending***

**12.1.1.49.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#DebbugsStatepending](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#DebbugsStatepending)**

- [Thing](#)
  - [State](#)
    - [NonTerminalState](#)
      - *DebbugsStatepending*

#### **Annotations**

- comment: "Complements the NonTerminalState class by adding debbugs specific subclass."

#### **12.1.1.49.2 Superclasses**

- [NonTerminalState](#)
- 

#### **12.1.1.50 Class: *DebbugsStatepending-fixed***

**12.1.1.50.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#DebbugsStatepending-fixed](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#DebbugsStatepending-fixed)**

- [Thing](#)
  - [State](#)
    - [NonTerminalState](#)



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence : LI-15-RAP
Version 1.0 du 16/09/09

- DebbugsStatepending-fixed

### **Annotations**

- comment: "Complements the NonTerminalState class by adding debbugs specific subclass."

#### **12.1.1.50.2 Superclasses**

- [NonTerminalState](#)
- 

#### **12.1.1.51 Class: Issue**

##### **12.1.1.51.1 <http://www.ifi.uzh.ch/ddis/evoont/2008/11/bom#Issue>**

- [Thing](#)
  - Issue
    - [BugzillaDefect](#)
    - [BugzillaEnhancement](#)
    - [DebbugsDefect](#)
    - [DebbugsEnhancement](#)
    - Defect
    - Enhancement

### **Usage**

- [isDuplicateOf](#) range Issue
  - [inBugtracker](#) domain Issue
  - [isIssueOf](#) domain Issue
  - [isMergedInto](#) domain Issue
  - [isMergedInto](#) range Issue
  - [hasSubIssue](#) domain Issue
  - [hasSubIssue](#) range Issue
  - [reportedAlsoIn](#) range Issue
  - [reportedAlsoIn](#) domain Issue
  - [isDuplicateOf](#) domain Issue
  - [hasIssue](#) range Issue
  - [isInProduct](#) domain Issue
- 

#### **12.1.1.52 Class: NonTerminalState**

##### **12.1.1.52.1 <http://www.w3.org/2005/01/wf/flow#NonTerminalState>**

- [Thing](#)





Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie

Référence : LI-15-RAP

Version 1.0 du 16/09/09

- [State](#)
  - NonTerminalState
    - [BugzillaStateASSIGNED](#)
    - [BugzillaStateNEW](#)
    - [BugzillaStateREOPENED](#)
    - [BugzillaStateRESOLVED](#)
    - [BugzillaStateUNCONFIRMED](#)
    - [BugzillaStateVERIFIED](#)
    - [DebbugsStatefixed](#)
    - [DebbugsStateforwarded](#)
    - [DebbugsStatepending](#)
    - [DebbugsStatepending-fixed](#)
    - Open
    - Resolved
    - Verified

### **Annotations**

- comment: "Fixed NonTerminalState of flow, to make it a subclass of State" (en)

### **12.1.1.52.2 Superclasses**

- [State](#)
- 

### **12.1.1.53 Class: Priority**

#### **12.1.1.53.1 <http://www.ifi.uzh.ch/ddis/evoont/2008/11/bom#Priority>**

- [Thing](#)
    - Priority
      - [BugzillaPriorityP1](#)
      - [BugzillaPriorityP2](#)
      - [BugzillaPriorityP3](#)
      - [BugzillaPriorityP4](#)
      - [BugzillaPriorityP5](#)
      - P1
      - P2
      - P3
      - P4
      - P5
-



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie

Référence : LI-15-RAP

Version 1.0 du 16/09/09

#### **12.1.1.54 Class: Product**

##### **12.1.1.54.1 <http://www.ifi.uzh.ch/ddis/evoont/2008/11/bom#Product>**

- [Thing](#)
  - Product

#### **Usage**

- [isInProduct](#) range Product
- 

#### **12.1.1.55 Class: Project**

##### **12.1.1.55.1 <http://usefulinc.com/ns/doap#Project>**

- [Thing](#)
  - Project

#### **Usage**

- [bugtracker](#) domain Project
- 

#### **12.1.1.56 Class: Resolution**

##### **12.1.1.56.1 <http://www.ifi.uzh.ch/ddis/evoont/2008/11/bom#Resolution>**

- [Thing](#)
    - Resolution
      - [BugzillaResolutionDUPLICATE](#)
      - [BugzillaResolutionFIXED](#)
      - [BugzillaResolutionINVALID](#)
      - [BugzillaResolutionMOVED](#)
      - [BugzillaResolutionWONTFIX](#)
      - [BugzillaResolutionWORKSFORME](#)
      - Duplicate
      - Fixed
      - Invalid
      - Later
      - Remind
      - ThirdParty
      - WontFix
      - WorksForMe
-



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie

Référence : LI-15-RAP

Version 1.0 du 16/09/09

### **12.1.1.57 Class: Severity**

#### **12.1.1.57.1 <http://www.ifi.uzh.ch/ddis/evoont/2008/11/bom#Severity>**

- [Thing](#)
  - Severity
    - Blocker
    - [BugzillaSeverityblocker](#)
    - [BugzillaSeveritycritical](#)
    - [BugzillaSeverityenhancement](#)
    - [BugzillaSeveritymajor](#)
    - [BugzillaSeverityminor](#)
    - [BugzillaSeveritynormal](#)
    - [BugzillaSeveritytrivial](#)
    - Critical
    - [DebugsSeveritycritical](#)
    - [DebugsSeveritygrave](#)
    - [DebugsSeverityimportant](#)
    - [DebugsSeverityminor](#)
    - [DebugsSeveritynormal](#)
    - [DebugsSeverityserious](#)
    - [DebugsSeveritywishlist](#)
    - Feature
    - Major
    - Minor
    - Trivial

---

### **12.1.1.58 Class: State**

#### **12.1.1.58.1 <http://www.w3.org/2005/01/wf/flow#State>**

- [Thing](#)
  - State
    - [NonTerminalState](#) +
    - [TerminalState](#) +

---

### **12.1.1.59 Class: TerminalState**

#### **12.1.1.59.1 <http://www.w3.org/2005/01/wf/flow#TerminalState>**

- [Thing](#)
  - TerminalState
    - [BugzillaStateCLOSED](#)



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie

Référence : LI-15-RAP

Version 1.0 du 16/09/09

- Closed
  - [DebbugsStatedone](#)
- 

### **12.1.1.60 Class: Version**

#### **12.1.1.60.1 <http://www.ifi.uzh.ch/ddis/evoont/2008/11/bom#Version>**

- [Thing](#)
  - Version

#### **Usage**

- [bugtrackerVersion](#) range Version
- 

#### **12.1.1.60.2 [helios\\_bt: objectproperties \(12\)](#)**

- [bugtracker](#)
  - [bugtrackerVersion](#)
  - [hasIssue](#)
  - [hasSeverity](#)
  - [hasSubIssue](#)
  - [inBugtracker](#)
  - [isDuplicateOf](#)
  - [isInProduct](#)
  - [isIssueOf](#)
  - [isMergedInto](#)
  - [reportedAlsoIn](#)
  - [reportedUpstreamIn](#)
- 

### **12.1.1.61 Object Property: bugtracker**

#### **12.1.1.61.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#bugtracker](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#bugtracker)**

#### **12.1.1.61.2 Annotations**

- comment: "A bugtracker of a project"

#### **12.1.1.61.3 Domains**

- [Project](#)



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie

Référence : LI-15-RAP

Version 1.0 du 16/09/09

#### **12.1.1.61.4 Ranges**

- [BugTracker](#)
- 

#### **12.1.1.62 Object Property: *bugtrackerVersion***

**12.1.1.62.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#bugtrackerVersion](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#bugtrackerVersion)**

#### **12.1.1.62.2 Annotations**

- comment: "The version of the bugtracker"

#### **12.1.1.62.3 Domains**

- [BugTracker](#)

#### **12.1.1.62.4 Ranges**

- [Version](#)
- 

#### **12.1.1.63 Object Property: *hasIssue***

**12.1.1.63.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#hasIssue](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#hasIssue)**

#### **12.1.1.63.2 Annotations**

- comment: "Adding the inverse link between bom:Issue and bom:Component, which was described in the evoont diagram."

#### **12.1.1.63.3 Domains**

- [Component](#)

#### **12.1.1.63.4 Ranges**

- [Issue](#)

#### **12.1.1.63.5 Inverses**

- [isIssueOf](#)



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence : LI-15-RAP
Version 1.0 du 16/09/09

#### **12.1.1.63.6 Usage**

- hasIssue inverse [isIssueOf](#)
- 

#### **12.1.1.64 Object Property: hasSeverity**

##### **12.1.1.64.1 <http://www.ifi.uzh.ch/ddis/evoont/2008/11/bom#hasSeverity>**

##### **12.1.1.64.2 Usage**

- [DebbugsEnhancement](#) -> hasSeverity only [DebbugsSeveritywishlist](#)
  - [BugzillaDefect](#) -> hasSeverity only ( [BugzillaSeverityblocker](#) or [BugzillaSeveritycritical](#) or [BugzillaSeveritymajor](#) or [BugzillaSeverityminor](#) or [BugzillaSeveritynormal](#) or [BugzillaSeveritytrivial](#) )
  - [DebbugsDefect](#) -> hasSeverity only ( [DebbugsSeveritycritical](#) or [DebbugsSeveritygrave](#) or [DebbugsSeverityimportant](#) or [DebbugsSeverityminor](#) or [DebbugsSeveritynormal](#) or [DebbugsSeverityserious](#) )
  - [BugzillaEnhancement](#) -> hasSeverity only [BugzillaSeverityenhancement](#)
- 

#### **12.1.1.65 Object Property: hasSubIssue**

##### **12.1.1.65.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#hasSubIssue](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#hasSubIssue)**

##### **12.1.1.65.2 Annotations**

- comment: "An Issue/Task can have subIssues / subTasks, in particular to model the Bug - Task relations in LaunchPad"

##### **12.1.1.65.3 Domains**

- [Issue](#)

##### **12.1.1.65.4 Ranges**

- [Issue](#)
-



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie

Référence : LI-15-RAP

Version 1.0 du 16/09/09

### **12.1.1.66 Object Property: *inBugtracker***

**12.1.1.66.1** [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#inBugtracker](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#inBugtracker)

#### **12.1.1.66.2 Annotations**

- comment: "An issue is present in a bugtracker"

#### **12.1.1.66.3 Domains**

- [Issue](#)

#### **12.1.1.66.4 Ranges**

- [BugTracker](#)
- 

### **12.1.1.67 Object Property: *isDuplicateOf***

**12.1.1.67.1** [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#isDuplicateOf](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#isDuplicateOf)

#### **12.1.1.67.2 Annotations**

- comment: "An Issue is a duplicate of another Issue, and it should be "closed", only having its duplicate stay open. In principle, there should be a restriction here on the state and resolution of that Issue (resolution == duplicate)"

#### **12.1.1.67.3 Domains**

- [Issue](#)

#### **12.1.1.67.4 Ranges**

- [Issue](#)

#### **12.1.1.67.5 Equivalentents**

- [isMergedInto](#)

#### **12.1.1.67.6 Usage**

- isDuplicateOf [isMergedInto](#)
-



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie

Référence : LI-15-RAP

Version 1.0 du 16/09/09

### **12.1.1.68 Object Property: *isInProduct***

**12.1.1.68.1** [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#isInProduct](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#isInProduct)

#### **12.1.1.68.2 Annotations**

- comment: "An issue is present in a product"

#### **12.1.1.68.3 Domains**

- [Issue](#)

#### **12.1.1.68.4 Ranges**

- [Product](#)
- 

### **12.1.1.69 Object Property: *isIssueOf***

**12.1.1.69.1** <http://www.ifi.uzh.ch/ddis/evoont/2008/11/bom#isIssueOf>

#### **12.1.1.69.2 Domains**

- [Issue](#)

#### **12.1.1.69.3 Ranges**

- [Component](#)

#### **12.1.1.69.4 Inverses**

- [hasIssue](#)

#### **12.1.1.69.5 Usage**

- [hasIssue](#) inverse isIssueOf
-





Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie

Référence : LI-15-RAP

Version 1.0 du 16/09/09

### **12.1.1.70 Object Property: *isMergedInto***

**12.1.1.70.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#isMergedInto](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#isMergedInto)**

#### **12.1.1.70.2 Annotations**

- comment: "An Issue has been merged into another Issue, and it should be "closed", only the merged issue should stay open. In principle, there should be a restriction here on the state and resolution of that Issue (resolution == merged)"

#### **12.1.1.70.3 Domains**

- [Issue](#)

#### **12.1.1.70.4 Ranges**

- [Issue](#)

#### **12.1.1.70.5 Equivalentents**

- [isDuplicateOf](#)

#### **12.1.1.70.6 Usage**

- [isDuplicateOf](#) *isMergedInto*
- 

### **12.1.1.71 Object Property: *reportedAlsoIn***

**12.1.1.71.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#reportedAlsoIn](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#reportedAlsoIn)**

#### **12.1.1.71.2 Annotations**

- comment: "An Issue can be reported in many different projects / distributions (see LaunchPad's also affects relations)"

#### **12.1.1.71.3 Domains**

- [Issue](#)

#### **12.1.1.71.4 Ranges**

- [Issue](#)



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence : LI-15-RAP
Version 1.0 du 16/09/09

#### **12.1.1.71.5 Usage**

- [reportedUpstreamIn](#) -> reportedAlsoIn
- 

#### **12.1.1.72 Object Property: reportedUpstreamIn**

**12.1.1.72.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#reportedUpstreamIn](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#reportedUpstreamIn)**

#### **12.1.1.72.2 Annotations**

- comment: "A particular case of reportedAlsoIn relation when it affects bugs in a distribution package and its corresponding upstream project's bug (see forwarded-to link in debugs)"

#### **12.1.1.72.3 Known Supers**

- [reportedAlsoIn](#)

#### **12.1.1.72.4 Usage**

- reportedUpstreamIn -> [reportedAlsoIn](#)
- 

#### **12.1.1.72.5 helios\_bt: dataproperties (1)**

- [bugTrackerURL](#)
- 

#### **12.1.1.73 Data Property: bugTrackerURL**

**12.1.1.73.1 [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/2009/07/helios\\_bt.owl#bugTrackerURL](http://picoforge.int-evry.fr/projects/svn/helios_wp3/2009/07/helios_bt.owl#bugTrackerURL)**

#### **12.1.1.73.2 Annotations**

- comment: "A string value to keep the URL of the bugtracker"

#### **12.1.1.73.3 Domains**

- [BugTracker](#)

#### **12.1.1.73.4 Ranges**


- string



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie

Référence : LI-15-RAP

Version 1.0 du 16/09/09

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

## **ANNEX : « Weaving a Semantic Web across OSS repositories: a spotlight on bts-link, UDD, SWIM »**

Nous reprenons ici un article présenté au colloque WOPDASD 2009 en juin 2009



Titre du document :	Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence :	LI-15-RAP
Version	1.0 du 16/09/09

## 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.vlascean  
u  
@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.

### Introduction

The HELIOS project<sup>20</sup> 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)

<sup>20</sup> <http://helios-platform.org/>



Titre du document :	Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence :	LI-15-RAP
Version	1.0 du 16/09/09

open source platform.

Among other goals, HELIOS aims at addressing bugtracker synchronization issues<sup>21</sup>. 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 *Semantic 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].

<sup>21</sup> A more detailed description can be found at : [https://picoforge.int-evry.fr/cgi-bin/twiki/view/Helios\\_wp3/Web/](https://picoforge.int-evry.fr/cgi-bin/twiki/view/Helios_wp3/Web/)

## 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.

### 12.2 *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



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie

Référence : LI-15-RAP

Version 1.0 du 16/09/09

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.

### 12.3 Ultimate Debian Database (UDD)

Some Debian Developers<sup>22</sup> have developed a

<sup>22</sup> Lucas Nussbaum, Stefano Zacchiroli and Marc Brockschmidt supervising the development made by Christian von Essen.

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<sup>23</sup> 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

<sup>23</sup> <http://melquiades.flossmetrics.org/>



Titre du document :	Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence :	LI-15-RAP
Version	1.0 du 16/09/09

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 rows across several tables with good performance<sup>24</sup>, while the field of native triple databases is still in its infancy.

## 12.4 Semantic Web enabled Issue Manager (SWIM)

*Semantic Web enabled Issue Manager (SWIM)* is a recently rolled-out application<sup>25</sup> developed as a follow-up to the Nepomuk<sup>26</sup> project. It stems from the fact that, as stated by Henry Story<sup>27</sup>:

*“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

24 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.

25 <http://club.mandriva.com/xwiki/bin/view/swim/>

26 <http://nepomuk.semanticdesktop.org>

27 Semantic Web evangelist at Sun Microsystems

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<sup>28</sup> 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.

28 <http://www.ifi.uzh.ch/ddis/evo/>





Titre du document :	Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence :	LI-15-RAP
Version	1.0 du 16/09/09

## 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.).

## 12.5 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<sup>29</sup> [9]. The work will focus

<sup>29</sup> the creator of the Ubuntu Linux project

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 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<sup>30</sup> 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.

## 12.6 Increased usability and data interchange

The use of Semantic Web techniques will

<sup>30</sup> <http://code.google.com/p/baetle/>



Titre du document :	Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence :	LI-15-RAP
Version	1.0 du 16/09/09

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 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 the 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.

## 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.

## 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](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/](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.

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

- [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] Damjanovic, 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



Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
Référence : LI-15-RAP
Version 1.0 du 16/09/09

## ANNEX : « Bugtrackers triplification »

Nous reprenons ici un court article proposé pour le challenge triplify 2009

### Bugtrackers triplification

Olivier Berger [olivier.berger@it-sudparis.eu](mailto:olivier.berger@it-sudparis.eu)  
Andrei Sambra [andrei.sambra@telecom-sudparis.eu](mailto:andrei.sambra@telecom-sudparis.eu)  
Valentin Vlasceanu ion [valentin.vlasceanu@it-sudparis.eu](mailto:valentin.vlasceanu@it-sudparis.eu)

(INF / TELECOM SudParis / Institut TELECOM, France)

Abstract: To interconnect bugtrackers, and especially the one used to manage free software projects, one need tools to convert their custom format to a common interoperable form. We, in the context of the Helios project, are working on refining existing ontologies to describe bugs from the most used bugtrackers in open source software. We propose two prototypes for review, based on triplify and EvoOnt BOM, which export bugs from bugzilla installations and Debian's UDD in the form of RDF triples.

Key Words: RDF, bug, bugtracker, triplify, Debian, bugzilla, UDD, Linked Data

#### 12.7 Introduction

Bugtracking is a vital activity for open source<sup>31</sup> development. Most open source software development groups use one bugtracker (among popular Open Source bugtrackers are bugzilla, Trac, Mantis, etc.). We, in the context of the HELIOS project[Helios Project], aim at addressing bugtracker synchronization issues.


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 is experimented.

#### 12.8 Bug tracking on the Semantic Web

For the needs of open source software maintainers and packagers, we aim at helping automating some of their daily work like monitoring the status of linked bugs. This will be helpful to free

---

<sup>31</sup>we consider free software and open source as synonyms in this document

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

software distribution maintainers, who package the software released by upstream projects, and maintain a link between users of the distributions (who have reported their problems to the distribution's bugtracker) and the upstream project's bugtracker.

To do so, it's essential to increase communication mechanisms, beyond bugtracker silos. Unfortunately, at the moment, such links between bugs are not always formally defined in the bugtracker applications, and their semantics is most of the times ambiguous. This used to render almost impossible the exploitation of such links without ad-hoc developments.

### **12.8.1.1 Bug tracker interoperability**

There are very few tools that allow to interact between bugtrackers. Launchpad is such a SAAS tool, but with an internal database, which doesn't allow use of its bugs model by external applications. Bts-link or Mylyn are other client tools, which do web-scraping or SOAP invocations, and both have custom code to address each bugtracker specific bug model.

### **12.8.1.2 Bug/Issue semantic representation**

The first step we have taken in Helios is to try and find an existing standard that can model bugs from the following bug trackers: Debian's Debbugs, Bugzilla, Mantis, Launchpad, Gforge/FusionForge, Trac. There's none, to our knowledge, established as such standard at the moment.

We are thus participating to the baetle[baetle] community that seeks to establish a standard for interoperability between bugtrackers, and are currently evaluating the use of EvoOnt BOM[EvoOnt], which we will extended, in the context of the Helios project<sup>32</sup>.

## **12.9 Current status**

We have developped two prototypes using Triplify to generate data comming from a copy of the Ultimate Debian Database[UDD] (which contains facts about Debian bugs[Debian]), and from any Bugzilla[Bugzilla] installation. Both will exhibit information in the form of RDF+N3 data. They should help as a reference implementation in order to validate the ontologies, and to start populating the Semantic Web with descriptions of Open Source projects Linked Bugs.

### **12.9.1.1 Bugzilla Triplification**


The triplify configuration we propose will expose facts about Bugzilla bugs (using EvoOnt BOM's Issue class as a base), the bugtracker's users (bug reporters, assignee, etc.) using SIOC's User and FOAF's Person, and the links between these : bugs reported by one account owned by a person, etc.

The detailed description, an example of the output data, and a link to a live demonstrator as well as the latest version of the config.inc.php file can be found on our project's wiki page at this URL [TriplifyBugzillaToRdf].

Please note that this is a work in progress implementation with a lot more features planned for the

---

<sup>32</sup>The BOM extension we are developing is available at [Helios BT Ontology], and is an ongoing work.

	Titre du document : Document de spécification : Modèle générique de suivi d'anomalie et interfaces Web Services du suivi d'anomalie
	Référence : LI-15-RAP
	Version 1.0 du 16/09/09

future releases (like modeling of the bug comments as SIOC discussions).

### **12.9.1.2 Ultimate Debian Database Triplification**

Much like the bugzilla triplification, we've designed a configuration of triplify [TriplifyUDDToRdf] using the same ontologies and relations to publish facts about Debian bugs, using the database of these bugs present in Debian's UDD[UDD].

We're running a demonstrator[UDD triplify demo] which provides real data relating to the hundreds of thousands of Debian bugs as well as their thousands of reporters (for a snapshot of UDD taken late July).

Particularly interesting are links between bugs, like the ones of Debian that are referring to bugs in upstream project's bugtrackers. These are described by forwarded-upstream relations in Debbugs, that we've mapped to a new helios-bt:reportedAlsoIn property. See [TriplifyUDDToRdf] for links to examples of such bugs on the live demonstrator.

## **12.10 Conclusion**

We wish to propose the two prototypes developed on triplify and the Helios BT ontology as our submission to the LOD Triplification Challenge 2009.

These first prototypes show that EvoOnt BOM fits very well the modeling of bugtracker bugs in the context of the use for bugs tracking on the Semantic Web, provided that some extensions are designed. We believe this will help pave the way to more Linked Open Data for open source bug facts in the coming months.

## **12.11 References**

[baetle] Bug And Enhancement Tracking Language <http://code.google.com/p/baetle/>

[Bugzilla] Bugzilla home page <http://www.bugzilla.org/>

[Debian] Debian's bugtracker <http://bugs.debian.org/>

[EvoOnt] Software evolution ontology <http://www.ifi.uzh.ch/ddis/evo>

[Helios Project] Helios project home page <http://www.helios-platform.org/>

[Helios BT Ontology] Helios Bugtracking Ontology [http://picoforge.int-evry.fr/projects/svn/helios\\_wp3/helios\\_bt.owl](http://picoforge.int-evry.fr/projects/svn/helios_wp3/helios_bt.owl)

[TriplifyBugzillaToRdf] Bugzilla Triplification wiki [https://picoforge.int-evry.fr/cgi-bin/twiki/view/Helios\\_wp3/Web/TriplifyBugzillaToRdf](https://picoforge.int-evry.fr/cgi-bin/twiki/view/Helios_wp3/Web/TriplifyBugzillaToRdf)

[TriplifyUDDToRdf] Bugzilla Triplification wiki [https://picoforge.int-evry.fr/cgi-bin/twiki/view/Helios\\_wp3/Web/TriplifyUDDToRdf](https://picoforge.int-evry.fr/cgi-bin/twiki/view/Helios_wp3/Web/TriplifyUDDToRdf)

[UDD] Ultimate Debian Database [wiki.debian.org/UltimateDebianDatabase/](http://wiki.debian.org/UltimateDebianDatabase/)

[UDD triplify demo] <http://kilauea.int-evry.fr:8081/triplify/UDD/>