
Tracking community intelligence with Trac

BY ROB BAXTER AND NEIL CHUE HONG*

*Software Sustainability Institute, EPCC, University of Edinburgh,
Edinburgh EH9 3JZ, UK*

We report on experiences at the Software Sustainability Institute (SSI) in customizing and using the Trac system to provide a single platform for recording, managing and tracking a wide range of community interactions. We note the essential requirement of a lightweight, easy-to-use system for recording ‘community metadata’ and discuss the pros and cons of using Trac in this way for day-to-day operations within SSI, and more generally as a means to record and track interactions with a wide and potentially very large community.

Keywords: Trac; community intelligence; lightweight issue-tracking system; requirements gathering

1. Introduction

The notion of ‘community intelligence’—that is, information coming from a community that can be used to provide information about the views, needs and expectations of that community to those wishing to provide a service, and its collection, management and analysis—is one that has been studied in many different fields (see [1–3]).

There have been several recent initiatives to better capture community intelligence from the UK e-Science community. These have provided unique insights into the requirements, needs and inhibitors of researchers and, where the assessment is performed on an ongoing basis, a way of identifying whether the community is responding to these requirements.

This paper examines the suitability of using issue-tracking systems—in particular, the tools provided by Trac [4]—as a means for recording, searching, sorting, analysing, assessing and reporting community intelligence. This is done within the context of the Software Sustainability Institute (SSI), a distributed team that supports the enhancement of software development in research communities.

2. Background

Community intelligence has been defined [5] as:

the capacity of a community to:

- understand the conditions in and around it—and its relation to those conditions;

*Author for correspondence (n.chuehong@epcc.ed.ac.uk).

One contribution of 12 to a Theme Issue ‘e-Science: novel research, new science and enduring impact’.

- initiate and respond in creative, coherent, appropriate ways in the context of those conditions; and
- to learn from its successes, failures, internal diversity and dissonance, and outsiders.

High levels of community intelligence are indicated by a vibrant sustainable community in which a high quality of experience is present, even in the face of changing conditions.

Another definition of community intelligence [6] from another field is:

Community intelligence is information acquired either directly or indirectly from a community, that when analysed can be used to inform [organizational] interventions. The information can come from variety of sources, but it will inform [the organization] about the views, needs and expectations of a community and the risks and threats posed to it or by it, either in terms of internal or external issues.

In this case, the organization is the Police, but we can see that, by removing the specific references to policing, this definition applies equally well to any other community, including that of e-Science. Community intelligence is inherently ‘bottom-up’; it is provided from the community and importantly reflects the more subtle interactions between different elements of that community, which may not be evident from aggregated information, e.g. from anonymized surveys.

As defined in a recent UK e-Science All Hands Meeting workshop [7], our starting point is that a minimally adequate community intelligence activity should be able, at any point in time, to answer questions such as:

- What are the current needs that researchers have articulated and how is the community responding to these needs?
- Are they being met?
- What have service providers and technology developers been doing to engage new user communities?
- How have they tried to address barriers to uptake?

The use of community intelligence to benefit a community is dependent on the overall cohesion of that community, and often determined by whether there are a number of ‘providers’ operating services to the broader group. In the case of the work we present, the broader community refers to the group comprising the users (researchers), developers (researchers and software engineers) and providers (researchers, institutional information technology services, national services) of software used to support research in UK higher education. This community is important because, while there are significant differences within the different domains that it covers, it is treated as a single community by most of the providers, due in part to the UK’s e-Science programme.

In the context of the move towards European-wide communities of e-Science practice (e.g. the European Strategy Forum on Research Infrastructures (ESFRI) projects that develop roadmaps for supporting pan-European research infrastructures) and infrastructure provision and support (the European Grid Initiative), the importance of understanding how the requirements of similar members of different communities relate to each other at regional, national and international levels increases. Here, the differentiation is between the

requirements of, say, the digital humanities, with their requirements for analysis of widely distributed collections of digitized materials, and the physics community, where significant data generation is focused on a single experiment and distributed outwards from there to collaborating groups. The need arises to ensure that interventions are coordinated so as to avoid unnecessary replication and inconsistency. Support for specific communities and the provision of general training infrastructures and material are areas where wider international collaboration seems essential. Likewise, is there benefit from community intelligence activities spanning continents? Do methodologies and processes transfer to different national contexts? Are the findings still relevant across communities?

UK e-Science stakeholders of many different kinds already actively engage with their communities. The issue is to understand the most effective mechanisms for collecting, analysing and sharing community intelligence, as each of these processes takes significant amounts of effort. Previous work to capture researcher requirements and needs has been carried out in the UK, notably the UK e-Science road trip report (2004) [8], SUPER survey (2006–2007) [9] and the JISC Community Engagement (eIUS, ENGAGE and e-Uptake) projects (2007–2009) [10,11], as well as internationally, e.g. in the USA looking at the Globus (2007–2008) [12] and TeraGrid (2006–2008) [13,14] communities. These projects have typically captured information from semi-structured interviews recorded on paper or audio, and then transcribed the data into different digital formats: spreadsheets, databases or XML (eXtensible Markup Language) encoded files. Other methods include surveys, observational studies, workshops and direct mining of public information.

Where projects engage with a large number of members of the e-Research community, it is important that the materials obtained can be reused as much as possible. There are a number of things that we must ensure are done both technically and administratively to enable this. These include the creation of a schema by which the information can be annotated with relevant metadata.

Once materials have been collected through whichever means has been decided upon, then it is important that a process by which the information can be processed and reduced to a set of key findings is designed. The remainder of this paper will examine the usage of a particular piece of software to support this process of capture and analysis.

3. The Software Sustainability Institute (SSI)

The SSI [15] is a national facility for research software users and developers (from the desktop personal computer to high-performance computing (PC to HPC)) of all disciplines. It provides effort, support and guidance to ensure that researchers can continue to use their chosen software as a cornerstone of their research, particularly beyond the lifetime of its original funding cycle.

This has a number of benefits: it improves the *capacity* to do research, by allowing more people to use powerful software tools; it increases the *capability* to do research, by enabling researchers to build in new functionality; and it allows for better *reproducibility* of results. The SSI works with research groups within the

UK to improve key research software through the provision of: online materials (tutorials, guides, assessment wizards); consultative advice (software evaluation, development process, community engagement, dissemination); collaborative partnerships (usability, quality, maintainability, refactoring); bidding for larger joint projects; and engagement with the wider international community.

As part of this, it is very important for the SSI to be able to track information from the different research communities, in particular, not only to ensure that a good service is provided but also to allow the identification of trends, overlaps and gaps in requirements and issues coming from the community.

4. The Trac system

Trac is ‘an enhanced wiki and issue tracking system for software development projects’ [16]. Trac provides a low-barrier-to-uptake platform that can be used to support a number of different popular development processes and policies. It provides interfaces to standard source-code control systems (with particularly close integration with SUBVERSION), an integrated wiki and customizable reporting facilities.

Trac also allows wiki markup in issue descriptions and commit messages. Importantly for the way that the SSI uses it, Trac supports the creation of links and references between bugs, tasks, changesets, files and wiki pages, and the concept of milestones (both time-based and task-based). A timeline shows all current and past project events in order, making the acquisition of an overview of the project and tracking progress very easy. The roadmap shows the road ahead, listing the upcoming milestones.

Trac’s wiki markup notation is both simple and intuitive, making the process of creating a true information web as easy as possible. Tickets in the ticket system can be referred to by their number—‘#32’ or ‘ticket:32’ for example—and a hyperlink is automatically created to the relevant entry. Stored reports (queries over the ticket database) can be referred to by number as ‘{10}’ or ‘report:10’, and entire changesets from the SUBVERSION repository can be referenced as, for example, ‘r128’ or ‘changeset:128’.

Trac has attracted a large userbase, from small, open-source projects to large blue-chip firms, and an active developer community. The TracHacks website [17] provides a host of additional plugin components to allow administrators to tailor the basic system to reflect particular ways of working. Trac administrators confident in working in Python can customize the system directly themselves.

5. Use of Trac within the SSI

Trac is used as the main tool for project management within the SSI. The SUBVERSION repository is used to store copies of documents that are being worked on by the team, and the wiki is used to record useful information, particularly administrative processes: a ‘living’ team manual.

It is the use of the ticketing system that provides the most benefit to the project: actions from meetings, tasks from project plans, deliverables and events can all be recorded and cross referenced. The customizability of the Trac

ticket system has enabled us to reshape it to a useful extent, rebadging the *Component* field as *Activity*—covering project-specific activities to the catch-all ‘Consultancy’—and creating new ticket types that cover different ways of recording information. Nevertheless, Trac is not by definition a true customer relationship management (CRM) system, so some guidelines have been developed to make it useful for tracking community intelligence.

6. Tracking community intelligence

The principal benefits of using Trac to record and analyse community intelligence are its low overhead to creating a ticket, its ability to link tickets to people, other tasks and milestones, and its ability to use the markup functionality to create custom metadata.

Low overhead is crucial. Empirical evidence from within the SSI and the OMII-UK collaborations, and anecdotal evidence from any and every research discipline, suggests that the more complicated the process for creating useful metadata, the less is created. Managing research data of any kind is made considerably easier by the application of meaningful metadata, but cajoling researchers, data creators or others to generate that metadata is still notoriously difficult. Community intelligence data is no exception. At the SSI, we have attempted to deploy Trac to meet two important requirements of gathering and managing the data:

- Information about a community contact should be recorded by the person who makes the contact.
- The recording mechanism must be quick and easy.

Initially at SSI, a very structured approach was considered, which involved the creation of a wiki page for each person and project whom the SSI communicated with. This would allow the creation of pages that tracked all information about the interactions with a particular subject. The disadvantage to this approach is that the tools for doing this automatically are limited in Trac (whereas they are more extensive in CRM systems) and the time to update an entry is relatively high. Previous experience of using wikis for CRM-like activities on the OGSA-DAI (Open Grid Services Architecture Data Access and Integration) project had also shown that, despite the strong structuring, without automatic compliance checking, it was possible to mis-link or otherwise corrupt records.

The current approach being used by the SSI is more lightweight. A ticket is created for each new interaction, which contains a minimal set of information: the SSI lead contact, the collaborators, the collaborating institution and the projects involved (figure 1). Initially, this is defined within a custom activity called Consultancy, with the key contact information added to the ticket’s *Description* field. Here a modest amount of discipline is required to record the right information in what is a freeform text box, although Trac’s particularly lightweight wiki markup at least does not hinder the process. The Trac markup engine will automatically hyperlink email addresses (to `mailto:` Universal

Create New Ticket

Properties

Summary:

Reporter:

Description:

Type: Priority:

Milestone: Component:

Keywords: Cc:

Blocked By: Blocking:

Start (YYYY/MM/DD): End (YYYY/MM/DD):

% Complete:

Assign to:

#118 (J. Smith: Cool Project review) ...

Software Sustainability Institute

logged in as rmb | Logout | Preferences | Help/Guide | About Trac

Wiki | Timeline | Roadmap | Browse Source | **View Tickets** | New Ticket | Search | Calendar

← Previous Ticket | Next Ticket →

Ticket #118 (new task)

J. Smith: Cool Project review Opened 32 seconds ago

Reported by:	rmb	Owed by:	rmb
Priority:	normal	Milestone:	M06
Component:	Consultancy	Keywords:	Cool Project
Cc:		Blocked By:	
Blocking:		Start (YYYY/MM/DD):	
End (YYYY/MM/DD):		% Complete:	0

Description

Contact(s):	John Smith j.smith@university.ac.uk	Reply
Institution(s):	The University	
Project(s):	The Cool Project	
Goal:	Review project website; suggest usability enhancements	
SSI Page:	CoolProject	

Figure 1. New ticket creation (before and after): minimal metadata for contact information is rendered automatically as appropriate hyperlinks. (Online version in colour.)

Resource Identifiers, URIs), web addresses (to [http:](http://) or [https:](https://) Universal Resource Locators, URLs) and words in ‘CamelCase’ to hyperlinks to internal wiki pages.

Figure 1 shows the actual Trac ticket creation interface for the following example.

```

|| Contact(s):      || John Smith j.smith@university.ac.uk ||
|| Institution(s): || The University ||
|| Project(s):     || [http://www.cool.uni.ac.uk/ The Cool Project] ||
|| Goal:          || Review project website; suggest usability enhancements ||
|| SSI Page:      || CoolProject ||

```

This ticket now provides a focus for any and all significant interactions with the contact. Notes, email exchanges, etc. can be pasted into the ticket's *Comment* field and the ticket updated to provide a history of the interaction, with each entry automatically timestamped. If additional tasks arise from a particular interaction, a new ticket can be created and linked back to the original. By default, Trac has no formal mechanism for linking tasks—creating links between tickets is a matter of using the standard internal notation of #N to create an automatic hyperlink to ticket number *N*. However, there are a number of plugins that can create formal dependence links between tickets, thereby establishing a dependence graph for particularly complex interactions. SSI makes use of the `MASTERTICKETSPLUGIN` to achieve this additional useful layer [18].

How often to create a new ticket rather than annotate an existing one is a matter of process and judgement rather than mechanics. This is where keeping the *Goal* of the original contact visible and in mind is useful—an interaction sufficiently outside the original scope almost certainly warrants a new ticket. Since the overhead of creating a new ticket is relatively low (especially since the ‘standard metadata’ from the *Description* field can—indeed should—be copied across), this is to be encouraged; it also provides a potential future metric for measuring the ‘weight’ of a particular interaction with a particular community or community contact: How many interactions have we had with John Smith this year?

7. Tracking collaborative projects

Within SSI, we make a distinction between short-term consultancy activities and longer-term development projects. From a management perspective, they are very different (in terms of resourcing, reporting, planning and so on), but from the perspective of collecting and managing community intelligence they are essentially the same. Thus, we have adopted a slightly unusual way of using the Trac system to record interactions within long-running development projects.

If consultancy discussions lead to a genuine collaboration, then we create a new, unique activity to represent the collaborative project—a new *Component* in the base system—and a special project report ticket, which is assigned as a ‘checkpoint’ rather than a ‘task’ (figure 2). The checkpoint ticket is then assigned to the SSI project leader and they use it as a place to record regular reports on progress—almost as a developer’s log. The fact that the project has its own *Component* means that other tickets—side discussions, project issues, etc.—can be created under its umbrella, and any existing tickets—such as the original Consultancy ticket that spawned the activity—can simply be reassigned with a couple of clicks. The reason for doing this is to allow Trac’s powerful reporting interface to be used to create custom reports that identify all currently active tasks, issues, etc. relating to an ongoing project.

The screenshot shows a Trac ticket interface for 'Ticket #45 (accepted checkpoint)'. The ticket title is 'Project report: SBIRC-I', which was opened 7 months ago and last modified 15 hours ago. The ticket is reported by 'rmb' and owned by 'mikej'. It has a 'normal' priority and is associated with the 'SBIRC-I' component. The milestone is 'M09'. The ticket is not blocked and has no blockers. The start date is '2010/08/30' and the end date is '2011/03/01'. The completion percentage is 40%.

The description, last modified by 'mikej', contains a table of project details:

Project Lead:	Mike J	Reply
Collaborators:	SBIRC	
	Paul Armitage < paul.armitage@ed.ac.uk > (Lecturer in Image Analysis)	
	Joanna Wardlaw < joanna.wardlaw@ed.ac.uk > (Director, Professor of Applied Neuroimaging)	
	Maria Valdes Hernandez < mvherman@staffmail.ed.ac.uk > (Research Fellow - Disconnected Mind)	
	Duncan Martin < dmartin2@staffmail.ed.ac.uk > (Scientific Business Manager)	
	Dominic Job < Dominic.Job@ed.ac.uk > (Data Manager)	
	David Rodriguez < david.rodriguez@ed.ac.uk > (SINAPSE)	
	Andrew Robson < arobson@staffmail.ed.ac.uk > (SINAPSE)	
SSI Effort:	3 months	

The ticket history shows a change by 'mikej' 2 months ago, including an attachment 'MVMxxxVI-Meeting-220910.TXT'. A subsequent change by 'mikej' 2 months ago describes a meeting with Maria, where the goal was to port ALE v1.2 to C++. The main concerns were the implementation of basic porting tasks:

- Reproduction of MATLAB matrix manipulation code in C++ (though the ops used are simple).
- Implementation of MVQ image quantization algorithm, though according to Wikipedia and implementations in other languages this is just a set of nested for loops. This is vital part of the tool.
- Implementation of ANALYZE header file reader. From the third-party GPL MATLAB code they have, this seems straightforward. The ANALYZE file spec. specifies the headers in detail.

The user plans to document these as part of WP3 and then decide how to proceed. A progress report was sent.

Figure 2. 'Checkpoint' tickets provide a single point of reporting for long-running collaborative projects. (Online version in colour.)

Other functions that become very simple to do are the creation of reports that identify all information connected with a particular person, project or institution. Tickets related to interviews can be tagged with keywords to represent issues or

requirements. Alternatively, if this has not been done, the reporting module is still able to search through the free text of the interview transcript where available. As additional information becomes available, e.g. through email follow-up, this can be added to the ticket and thus recorded.

By this approach, we have ‘evolved’ a system for recording community intelligence that fits very well into our other processes. In particular, it allows for the seamless migration of an information collection activity into a project that is undertaking software development into an ongoing collaboration, therefore creating a ‘full-lifecycle’ information tracking system. It seeks to strike a balance between ease of entry and quality of information.

Using Trac in this way places an emphasis on task and activity management—the natural strengths of the tool—ahead of contact or community relations. Capturing effective community intelligence still relies on a certain degree of discipline among the system’s users, for adding appropriate contact information to created tickets. One plugin that might provide a little more formality to recording contact information is the CLIENTSPPLUGIN [19]. While aimed primarily at software houses, this plugin provides for the creation of *Clients* as ‘top-level entities’ after the fashion of *Components*. While requiring more investment from a reporter recording a new contact—a new *Client* must be created through the administration interface—this approach would have the advantage of standardizing contact information by providing each ticket with a drop-down list of recorded *Clients* from which to assign. This would promote *Client* to a fully queryable field within the internal database and would greatly facilitate the tracking—if not necessarily the collection—of intelligence related to SSI’s many communities of interaction.

8. Tracking long-term engagement: an example

A key requirement of the sort of consultancy activity that SSI undertakes is to ensure that, at some point in time after the initial work is completed, a check-back is made with the original client. How have they got on? Have the recommendations proved useful? Has any other issue arisen that SSI could assist with? This sort of ‘account management’ activity is often one of the most challenging to keep track of, especially when trying to serve a large, diverse research community. This actual example from the SSI’s first six months of operation illustrates the approach we have adopted.

Through an existing contact, SSI received a request to perform a ‘sustainability evaluation’ of software from the Virtual Research Integration Collaboration (VRIC) project [20]. VRIC seeks to embed a virtual research environment into the Royal National Orthopaedic Hospital, and the long-term effectiveness and sustainability of the software platform is of paramount importance in a case like this.

The initial contact triggered the creation of a ‘new Consultancy task’ ticket in the Trac system. Discussions at SSI’s weekly review meeting identified an appropriate time window from the right consultant, and the task moved from ‘new’ to ‘assigned’, and on to ‘accepted’ when work on the software review began.

Comments added to the active ticket on a regular basis (typically daily) created a ‘mini-blog’ documenting activity on the review. After a week or so of effort, the evaluation report was delivered to the client and the ticket formally ‘closed’ at the weekly SSI review. However, a second ticket was immediately opened—a ‘follow on’ ticket set against a uniquely created *Milestone* in the Trac roadmap. This was linked back to the original Consultancy ticket using the dependence graphing functionality provided in the `MASTERTICKETSPLUGIN`, thereby creating the beginnings of a visual chain of interactions with that particular client.

This was ‘assigned’ to the original software consultant but left unaccepted; the ticket captures the requirements for a check-back with the original client at a suitable point in time (the unique *Milestone*), to see how work on the evaluation recommendations has progressed (or not), and whether any additional assistance from SSI could prove useful. Use of Trac in this way enables us not only to track project activity but also to reflect the aspects of account management we need effectively to track our impact and usefulness to the research community.

This shows the ability of Trac to be used in a way that can deal with the often sporadic nature of long-term community intelligence data gathering: by enabling not only associations to be recorded, but changes in the nature of the engagement to be illustrated and allowing actions required in the future to be noted.

9. Future work

A future use of the system will be to better understand trends; because dates are associated with all information records, it should be possible to create timelines associated with different requirements. Depending on the nature of the reports required, it may prove necessary to create additional Trac plugin components in Python to work directly with the underlying database.

Another consideration is to understand how to manage the process of exposing the information collected more widely. Trac allows for authorization of authenticated users of the system; however, this is applied to tickets on a per-ticket basis, rather than at the level of individual comments within a ticket. If information is to be shared between different organizations using separate Trac instances, there must be an appropriate means of exposing and linking, synchronizing, or exporting and importing tickets. In this respect, it is similar to the requirement to link bugs between separate issue trackers in dependent software projects.

Ultimately, we look to create communities of practice both within the different research communities the SSI works with—to improve the sustainability of the software developed and used by them—and across the different organizations tasked with providing useful interventions as a whole.

A community of practice is a group of people, who share similar challenges, interact regularly, learn from and with each other to improve their ability to address their challenges.

(Etienne Wenger)

In so doing, we can liberate the creative potential of the members of the community, grow new individual and collective capabilities, and introduce new ways of working that will benefit the whole community.

10. Other approaches

There are other commonly used ticketing and issue-tracking systems available, including JIRA, Bugzilla and RT. Bugzilla is commonly used by open-source projects to track tasks, and we are aware of its use by the Globus Project as a mechanism for automatically defining roadmaps [21]. However, we are unaware of any groups who are using issue-tracking systems to easily allow the migration of community intelligence gathering tasks to development tasks.

Likewise there are many CRM systems, e.g. SugarCRM [22], which have more powerful CRM functions that may be useful for recording and analysing community intelligence. However, experience from OMII-UK has shown that these are expensive to customize for a research-oriented workflow.

An interesting method for the publishing of community metadata was undertaken by the e-Uptake project, which used the CONNEXIONS [23] platform for open documentation to record their findings [24].

11. Conclusions

We have presented our approach to using a common issue-tracking and project management tool, Trac, to better record and analyse community intelligence gathered as part of the SSI's activities. The key to the successful use of the system has been to combine Trac's powerful markup, linking and reporting functionality with a low-overhead process for adding minimal amounts of required metadata. Overall, the benefit of using such a system in this way is the ability to track information as the nature of the interaction with the community changes. We believe that this approach may also be of benefit to other organizations acting as providers to the broader research community and we shall seek to identify in the future if more efficiency can be found in sharing community intelligence enabled by the systems we use to record the information.

The Software Sustainability Institute, including the work described in §8 and shown in figure 2, is funded by EPSRC grant EP/H043160/1. We acknowledge the experience and input from colleagues on the OMII-UK, OGSA-DAI, e-Uptake and ENGAGE projects for their contributions of information about their use of alternative processes for recording community information. We also acknowledge the contributors to the workshop on 'Gathering Requirements for Future e/Cyber-infrastructure' held in Chapel Hill, NC, in May 2009, which provides some of the background to this paper.

References

- 1 Huss III, J. W., Lindenbaum, P., Martone, M., Roberts, D., Pizarro, A., Valafar, F., Hogenesch, J. B. & Su, A. I. 2010 The Gene Wiki: community intelligence applied to human gene annotation. *Nucleic Acids Res.* **38**, D633–D639. (doi:10.1093/nar/gkp760)
- 2 Alonso, F., Walsh, C. O., Salvador-Carulla, L. & the eVITAL group. 2010 Methodology for the development of a taxonomy and toolkit to evaluate health-related habits and lifestyle (eVITAL). *BMC Res. Notes* **3**, 83. (doi:10.1186/1756-0500-3-83)
- 3 Lee, S.-Y. D., Chen, W. L. & Weiner, B. J. 2004 Communities and hospitals: social capital, community accountability, and service provision in U.S. community hospitals. *Health Serv. Res.* **39**, 1487–1508. (doi:10.1111/j.1475-6773.2004.00300.x)
- 4 Edgewall Software. 2010 Trac frequently asked questions. See <http://trac.edgewall.org/wiki/TracFaq>.

- 5 Atlee, T. 2004 Functions that make up community intelligence. Co-Intelligence Institute Report. See <http://www.co-intelligence.org/CommunityIntellFunctions.doc>.
- 6 Innes, M., Roberts, C. & Maltby, S. 2005 Community intelligence: a report to the National Community Tensions Team. London, UK: Association of Chief Police Officers.
- 7 Voss, A. 2010 Enhancing community intelligence for eScience (workshop). In *UK e-Science All Hands Meeting 2010. e-Science: novel research, new science and enduring impact*. See <http://www.allhands.org.uk/2010/session/w5>.
- 8 Schopf, J. M. & Newhouse, S. 2004 State of Grid users: 25 conversations with UK eScience groups. UK e-Science Technical Report Series, UKeS-2004-08, October 2004. See http://www.nesc.ac.uk/technical_papers/UKeS-2004-08.pdf.
- 9 Newhouse, S., Schopf, J. M., Richards, A. & Atkinson, M. P. 2007 Study of user priorities for e-Infrastructure for e-Research (SUPER). UK e-Science Technical Report Series, UKeS-2007-01, April 2007. See http://www.nesc.ac.uk/technical_papers/UKeS-2007-01.pdf.
- 10 Hardisty, A., Chue Hong, N., Procter, R., Hughes, L., Voss, A. & Fraser, M. 2009 Birds of a Feather (BoF) session on 'e-Infrastructure: tool for the elite or tool for everybody?'. UK e-Science Technical Report Series, UKeS-2009-01, January 2009. See http://www.nesc.ac.uk/technical_papers/UKeS-2009-01.pdf.
- 11 Mascord, M., Argüello Casteleiro, M., Fraser, M., Voss, A., Procter, R., Halfpenny, P. & Jirotko, M. 2007 Scoping e-Infrastructure usage: interim report. November 2007. e-Infrastructure Use Cases and Service Usage Models (eius) Project. See <http://www.eius.ac.uk/scoping/eIUS-Scoping-einfrastructure.pdf>.
- 12 Childers, L., Liming, L. & Foster, I. 2008 Perspectives on distributed computing: thirty people, four user types, and the distributed computing user experience. Argonne National Laboratory, Technical Report ANL/MCS/CI-31, September 2008. See http://www.mcs.anl.gov/~childers/perspectives/ANL_MCS_CI_31_no-binding.pdf.
- 13 Zimmerman, A. & Finholt, T. 2008 Report from the TeraGrid evaluation study, Part 1: Project findings, August 2008. See <http://hdl.handle.net/2027.42/61838>.
- 14 Zimmerman, A., Krause, M., Lawrence, K. & Finholt, T. 2008 TeraGrid evaluation report, Part 2: Findings from the TeraGrid user survey, July 2008. See <http://hdl.handle.net/2027.42/61839>.
- 15 Software Sustainability Institute. 2010 Software Sustainability Institute website. See <http://www.software.ac.uk/>.
- 16 Edgewall Software. 2010 Trac Open Source Project website. See <http://trac.edgewall.org/>.
- 17 Renzmann, M. 2010 TracHacks website and repository. See <http://trac-hacks.org/>.
- 18 Kantrowitz, N. 2008 Trac MasterTicketsPlugin documentation. See <http://trac-hacks.org/wiki/MasterTicketsPlugin>.
- 19 Guthrie, C. 2008 Trac ClientsPlugin documentation. See <http://trac-hacks.org/wiki/ClientsPlugin>.
- 20 Wills, G., Gilbert, L., Grange, S. & Recio, A. 2010 VRIC—Virtual Research Integration Collaboration: a VRE aimed at creating a framework to manage the research life cycles of basic science and clinical research. University of Southampton Technical Report. See <http://eprints.ecs.soton.ac.uk/21990/>.
- 21 Globus Project. 2010 Globus Project roadmap items. See <https://bugzilla.mcs.anl.gov/globus/buglist.cgi?component=Roadmap>.
- 22 SugarCRM. See <http://www.sugarcrm.com/>.
- 23 Henry, G. 2004 Connexions: an alternative approach to publishing. In *Research and Advanced Technology for Digital Libraries, Proc. 8th European Conf., ECDL 2004, Bath, UK, 12–17 September* (eds R. Heery & L. Lyon). Lecture Notes in Computer Science, no. 3232, pp. 421–431. Berlin, Germany: Springer. (doi:10.1007/978-3-540-30230-8_38)
- 24 Voss, A. & Asgari-Targhi, M. 2009 e-Research community engagement findings, 9 June 2009. See <http://cnx.org/content/col10673/1.9/>.