Abstractions, Accounts and Grid Usability

M. Hartswood[1], Y-W. Lin[2], R. Procter[2], J. M. Schopf[3], R. Slack[1], J. Ure[1] and A. Voss[1]

[1] School of Informatics, University of Edinburgh

[2] National Centre for e-Social Science, University of Manchester

[3] National e-Science Centre, Edinburgh

Email address of corresponding author: rob.procter@ncess.ac.uk

Abstract. The vision of the Grid is one of seamless, virtual and constantly changing resources where users need not concern themselves about details, such as exactly where an application is running or where their data is being stored. However, seamless and virtual often imply a lack of control that users may be wary of, or even opposed to. Drawing upon our studies of HCI and of collaborative work, this paper examines whether the Grid development community should be taking this vision literally and argues for the need for accountability of systems 'in interaction'. We give examples of an alternative approach that seeks to provide ways in which administrators, technical support and user communities can make sense of the behaviour of the complex socio-technical ensembles that are the reality of Grids.

Introduction

In 'The Grid: Blueprint for a New Computing Infrastructure', Kesselman and Foster (1998) defined a computational Grid to be "a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities." This, in turn, has been likened to providing a utility service, such as electricity where a user doesn't specify whether their current comes from a nuclear power plant or wind farm[1], a Grid user will simply plug into their "Grid port" on the wall and have access to as much compute and data resource as needed. "This idea is analogous to electric power network (grid) where power generators are distributed, but the users are able to access electric power without bothering about the source of energy and its location."[2]

Currently, this vision is still far off: today's Grids do not reach this level of abstraction. In most cases, compute resources a user needs must be specified, data must be named by a physical file name and explicitly transferred, and the authorization and authentication framework needed to enable these interactions is quite lengthy and often onerous. Nevertheless, the Grid is often promoted as a seamless, virtual and dynamically changing resource, where users will have no need to concern themselves with details such as where their applications are running, who is providing the services they are using, exactly what state their data is in, or low level issues of control[3], and research to reach the goal of a completely virtual and seamless environment continues.

---

[1] Green consumers may want to know and, in some cases they are now being offered the choice.

[2] http://www.gridcomputing.com

[3] Except as specified in Service Level Agreements (SLAs). SLAs are …

Drawing upon our studies of Human-Computer Interaction and of collaborative work, we argue that it is not clear that seamlessness is what users want from the Grid: taking seamlessness literally means users must sacrifice the ability to exercise control (Hardian et al., 2006) and to make systems accountable 'in interaction'. We do not deny that users benefit from abstractions in that they protect users from unnecessary details. These abstractions must, however, work at differing levels, for different types of users and be queryable on demand so that users can understand system behaviour and determine appropriate courses of action should problems arise. This is what we refer to as providing accountability in interaction. As evidence, we give examples of how control and accountability in interaction is played out in similar environments. We then examine accountability issues as they arise in Grid environments, focusing in particular on Grid as an enabler of e-Research whose own vision is of distributed, large scale collaboration conducted through the agency of 'virtual organisations' (VOs) which determine policies for sharing resources between participating administrative domains. We conclude by considering ways in which accountability may be delivered in Grid environments, exploring how data about Grid and user behaviour can be mined, organised and visualised in appropriate ways to provide accountability to different user communities, i.e., VO administrators, Grid support staff and end users of different levels of sophistication.

Our approach is to examine how users of Grids generate data as a by product of their activities (i.e., system-system, system-user, user-system, user-user interactions) and explore how this data can be mined, organised and visualised in appropriate ways to provide accountability to different user communities, for example, Grid administrators, technical support staff and end users of different levels of sophistication. In this paper we will discuss how this might be done, giving examples of the kinds of data that can be mined and how we can utilise them to afford awareness and accountability. In addition, we will look at the ways in which people can use the resources so provided to furnish accounts of a system's workings and people's activities within a virtual organisation.

Accounts and Accountability: Initial Problem Statement

When we look at the kinds of information that systems provide of their operations we can, quite obviously, state that this falls well short of the kinds of accounts that we would expect our colleagues and friends to give if, for example, they were late to a meeting or similar. We must, therefore, say that what systems provide are not accounts but the resources for the assembly of accounts by humans – that what a system does is to furnish resources as opposed to accounts. Now that may seem to be being overly careful regarding the choice of terminology – indeed it may be seen by some as a terminological quibble of little consequence – yet we want to argue that the gap between the systems' provision of resources for assembling accounts and accounts of what the system is doing is key to our understanding of grid usability[4].

Claims that the Grid is a way of looking anew at data and computational resource-sharing are undermined if, to put it bluntly, we do not know what on earth the system is doing. How can we trust a system whose only indications that something may be awry either in terms of software or security is opaque for many users? The cognoscenti may suggest opening up

[4] There is, of course, an argument that systems provide something very close to accounts – a claim often heard in strong AI or similar circles. We would dispute that, despite their apparent proximity to accounts, these are not accounts in themselves but at best proxies for or simulations of accounts. We shall take this up below.

firewalls and ports and using this or that service or similar, but unless we know what the consequences of these actions might be we risk some unpleasant surprises[5].

To clarify this issue we shall use the following nomenclature: while what systems do is an accountable matter, but we shall refer to the information that systems provide as 'account resources[6]' and 'accounts' to refer to what humans do with these. Thus, if my system is running slowly that is an accountable matter (I want to know why) and I may query my system and obtain some indication of the issue (the account resource) which I will then mention in, for example, my discussion with technical support personnel and in deciding what is happening and what to do about it

In what follows we will look at the ways that systems provide information on what they do and how these can be used in the assembly of accounts.

The Web

The Web is often held up as a usability benchmark for the Grid. People point to the Web's transparency as one of the key reasons behind its global adoption and reason that the Grid must similarly succeed in having seemless use if it is to make the transition from a tool for technically able users to a tool for the average researcher. However, the assumptions about Web use which have inspired this aim do not bear close scrutiny. In the early days of the Web, especially, the abstraction of a seamless, distributed information space where the location of resources is irrelevant was rarely sustained in practice (Johnson, nnnn).

The globally distributed nature of Web resources and the character of the underlying internet can combine to create unpredictable delays as requested pages and files download. Users are observed to deal with this behaviour in various resourceful ways which call upon, for example, an understanding of how the technology behind the browser works or reasoning about the effects of temporal rhythms in internet use (Stanyer and Procter, 1999). It was common, for example, to observe users reloading a web page when progress was slow in the belief that a faster connection to the host server might thereby be obtained. While it might be true that getting a faster connection is not what happens, there are a number of possible reasons why hitting the reload button may solve the problem of request being stalled, e.g. getting a connection to a different server in a load-balanced environment. While the explanation of the system's behaviour may not be technically correct, the heuristic employed is suitable in the case of a short transaction like a request for a webpage. However, in the case of longer-running or more expensive transactions, simply restarting is not an option and hence there is a greater need for the system to provide more account resources. In these and other examples, we see how the behaviour of the Web becomes an 'accountable' matter in use, reflecting the importance of providing explanations for 'why things are this way' that enable users to deal with problems.

Web browser user interfaces have generally fallen short in regard to making download behaviour accountable, preferring to stick to the abstraction of a seamless information space and advances in network bandwidths, the use of 'mirror' sites, etc., may have rendered this aspect of Web behaviour irrelevant. However, as the Web is put to new uses, then new accountability issues are now beginning to emerge, some of which are still technical in nature, while others reflect the Web's evolution into a populated, collaborative, social space

---

[5] This is not to be alarmist – what is at stake ranges *inter alia* from significant security issues to a system not working reliably.

[6] For want of a better term. We shall also have cause to distinguish between a number of 'levels' of these.

in which people do business, interact and strike up relationships. These new accountability issues reflect users need to have answers to questions about relevance, quality, security, identity and trust: does this link take me to a Web page with the information I'm looking for? Is the content of sufficient quality? Is it safe to send my credit card details? How can I verify the identity of a Web site or an individual? How can I tell if this email is really from my bank or if this download is what it claims to be?

All of these questions make for the Web becoming less rather than more seamless and have led to the devising of new ways to make the experience of using it more accountable. Google's success in the search engine market owes a lot to its approach to the relevance question as applied to the ordering of search hits, one which is simple enough for most users to understand; Amazon, like many e-commerce sites, uses purchaser ratings[7] to advise potential customers of what products to buy; eBay provides information to potential buyers about sellers' track record in completing deals and recently bought the Voice Over IP company Skype allegedly so its buyers and sellers can talk to each other via their computers.[8]

The Grid

The vision of Grid computing is one of building complex systems from dynamically assembled individual, distributed resources, and bridging across organisational and technical boundaries. Like the Web, Grid users are presented with an abstraction of seamless access to resources, but to distributed computation and services as well as distributed data. In the Grid vision, these resources may be composed to carry out complex functions on behalf of users, without users having to know where they are or how they are provided. Technically, a resource can be almost anything digital: a large-scale supercomputer, a computational service, a database or an instrument collecting in real time. In complex applications, numerous such resources get tied together to provide a particular function. Because these resources are defined in a machine readable form, in principle, Grid users need only provide a high-level description of what they require and middleware will find resources that match and compose them appropriately. It is the relative simplicity of these abstractions that enables Grid middleware to be widely applicable and support a vast range of applications.

Complex ensembles of diverse, distributed resources belonging to different administrative domains create challenges for infrastructure management and use: resources and services may change over time, resources may become temporarily or permanently unavailable; services may not behave in the expected fashion and causes may need to be diagnosed and fixed. Such events can lead to a number of difficulties for ensuring the infrastructure remains in 'working order'. For example, users might want to make judgements about when a service might resume; system administrators have the problem of understanding the relationships between observed behaviour and complex configurational data, Grid support staff need to diagnose and fix faults. In addition, resources are normally managed under local arrangements within the organisations that own them and will be subject to different policies and conditions of use. Furthermore, they will often be used by local as well as remote users, be part of multiple virtual organisations and may have more than one purpose (e.g., academic lab resources may be used for both course work and running simulations for research).

The Grid is not an end in itself but an enabler of the e-Research vision of large scale and collaborative scientific investigation through the agency of virtual organisations. This raises the question to what extent the workings of a virtual organisation (VO) need to be made

---

[7] Just one example of the use of social recommender techniques.

[8] http://news.bbc.co.uk/1/hi/business/4237338.stm

accountable to individuals involved in these collaborations. In conventional settings, collaboration is supported through the affordances of physical spaces. In her study of scientific practice in conventional research settings, Knorr-Cetina (1999) observes that "Laboratories are now collective units that encapsulate within them a traffic of substances, materials, equipment and observations … The traffic of objects, researchers and information produces a lifeworld within which laboratories are locales" (p.39). Physical places afford opportunities for collaborators to observe what's going on, remain informed of relevant developments and coordinate their activities. It is through such interactions that researchers are able to work up the trust that is essential for collaboration to be effective.

As with the Web, thinking of the Grid as supporting a populated, social and collaborative space raises many interesting issues for how these VOs may be represented so as to make activity within them observable and accountable, and thereby afford awareness and understanding among their members. We can think of VOs as not only binding together researchers, but also administrators and technical support staff collaborating to provide trustable, dependable and useful services (Bowers, 1994; Rogers, 1992) and share resources. We are interested in how VOs, as populated spaces, as ensembles of technologies and people, might be made tangible through collaborative affordances (ways of making available rules for involvement, making those involved accountable to those rules, and making those involved and their activities visible to other participants) and how operational information might be made usefully available to Grid administrators, technical support staff and end users. That is, in addition to monitoring the status of the various technical resources that underpin a VO, there is a question of users maintaining an awareness of what others are doing in the context of the VO and how it relates to their own work.

If VOs are to support collaboration effectively, then we need to consider how Grids might afford not only interaction in the context of environments designed explicitly for collaboration[9] but also how it can serve as a resource for researchers maintaining awareness of relevant events happening within the VO of which they are members. Accountability issues that arise might include trusting remote sites to run applications or to host data securely. For example, de Paul et al. (2005) have been examining how people's sharing of data can be made accountable through visual representations of file use. Our recent research suggests that there are a number of issues for accountability regarding data sharing in e-Science – e.g., data quality and confidentiality – which we need to understand and to provide affordances for in distributed research environments (Jirotka et al., 2005).

Account Resources and Accountability: Some Examples

In this section we examine examples of various attempts to provide account resources to explain system behaviour, including ways in which the behaviour of the Grid is currently monitored and how users can use this data to make operational decisions.

Firewalls

Firewalls provide an interesting example of how Web behaviour becomes an accountable issue and how current solutions may not always be satisfactory as warrants for users to 'trust' the system. The following example demonstrates the implementation of system security policy for potentially problematic Internet transactions (Anderson et al., 2003). In Figure 1, we see what we might call a 'level one' account resource of a situation in which the user is

[9] For example, Access Grids provide sophisticated video conference facilities for formal collaborative settings such as meetings. See http://www.accessgrid.org

called upon to make a decision: a firewall has produced an alert concerning an application's attempt to access the Internet. Clearly, access to the Internet for downloading or uploading data to/from the system can be an accountable matter for system security. The firewall's implementation of policy with respect to these transactions is statically configured and involves calling upon the user to decide if access requests are appropriate and can be proceeded with.

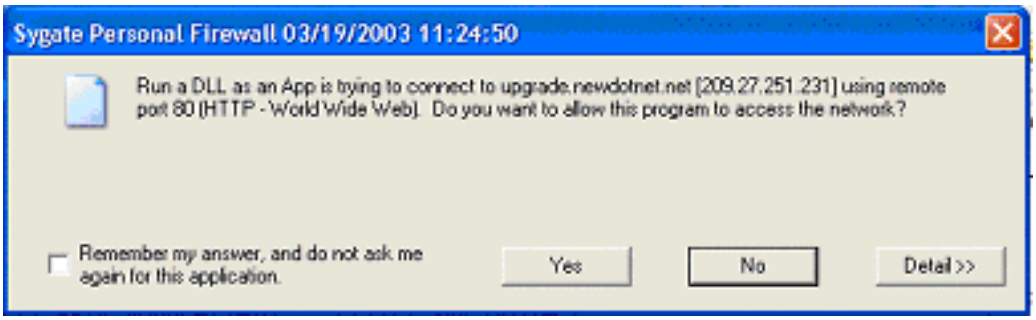

Figure 1: An example of a 'level one' account resource.

Figure 2 shows the 'level two' account resource produced in response to the user's request for more information. Note that the policy implementation requires user involvement since the firewall software cannot by itself determine if the transaction is to be viewed as a safe one. It is expected that the user has responsibility for deciding whether the transaction should be continued, and that the user has the requisite knowledge to decide whether the remote site is a trustworthy, whether the transaction is legitimate for this application and so on. Thus, implementation of the policy can be thought of as partial, requiring the user to 'fill in the gaps' on the occasion of its application.

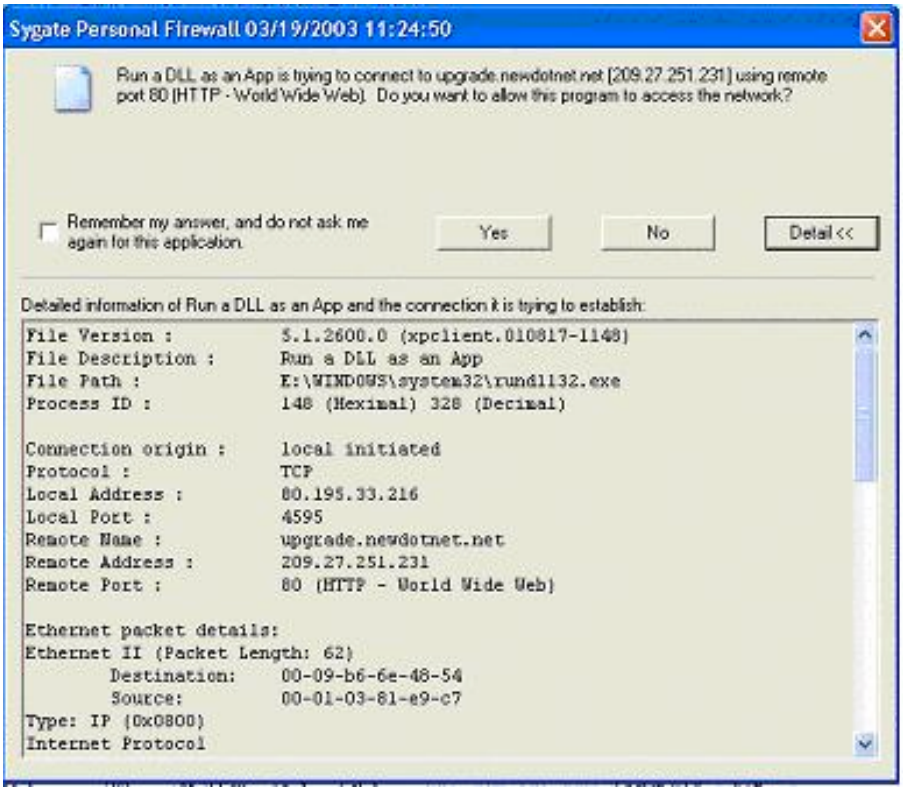

Figure 2: A 'level two' account resource derived from the level one account resource in Figure 1.

What the firewall system dialogues offer are account resources at two 'levels' of description. The 'level one' account resource informs that there is some action required and gives some basic details. The 'level two' account resource furnishes further details about the nature of the transaction at a protocol level. The level two account resource is intended to help a user reach a decision about whether or not to allow the transaction to proceed, but unless the user has prior knowledge of the trustworthiness or otherwise of the given www site or is able to spot irregularities in the details of the transaction, then such account resources will be of little use. The account resource omits, for example, a description of why it was necessary for the application to make this particular transaction at this particular time. An improved basis for decision-making might be afforded by the firewall accounting for the context of the transaction: has the message appeared in response to the user's action or due to some background process? Is this a legitimate transaction for this application at this time? What are the potential consequences of the transaction?

Grid Security

A prime example of the problem of providing adequate account resources is Grid security which is often realised through a public key infrastructure (PKI). On the one hand there is the need to reduce the complexity of managing trust relationships, which currently often involves manual handling of X.509 certificates (Beckles, Welch and Basney 2005). There are a number of problems associated with this, not least problems associated with keeping private keys secure[10], managing a PKI configuration, potentially on a large number of computers (e.g., then users are mobile and use multiple machines), and managing certificate revocation lists. There is some evidence that the complexity of managing a PKI leads some people to take shortcuts such as sharing certificates, raising serious security issues (e.g., Balfanz et al. 2004). Also, as Beckles, Welch and Basney (2005) point out, Grid services often 'fail closed':

"This tends to make grid security configurations rather brittle, as a small error can cause all security operations to fail. Combining this with users who are not (and do not wish to be) experts in grid security configurations results in a significant usability issue" (ibid., p. 78).

A number of approaches exist that provide a management environment that allows the complexities of the PKI to be hidden from end users by providing a façade that makes the Grid more easily usable under normal circumstances and leaves the complexity of the underlying Grid security infrastructure to be managed by Grid engineers and systems administration staff. Examples are PKIBoot (Gutmann 2003), MyProxy (Basney, Humphrey and Welch) or SACRED (Arsenault and Ferrell 2001, Gustafson, Just and Nystrom 2004).

The idea is to establish a 'single-sign-on' environment where a user can authenticate once using a simple username and password combination and then access any Grid resources they are authorised to use without needing to worry about the ways in which they are authenticated to remote systems. Grid portals are popular ways to providing the user interface for this façade. While we would applaud this development, there is the question what happens in those circumstances when, for one reason or another, circumstances are 'other than normal'. For example, Beckles, Welch and Basney point out that:

---

[10] Beckles, Welche and Basney (2005) describe the implications of poor visibility of arrangements on security, pointing out that users may accidentally store private keys on network storage since some systems make the use of networked storage space transparent (i.e. invisible) to the user. This opens up the possibility that security might be put at risk if no additional measures such as encryption of the key or the network filesystem traffic are used.

"Clear, robust error reporting is vital to making such an architecture viable, particularly since the underlying security protocols are unfamiliar to most users. If credentials are being generated ''on-the-fly'', then the time this takes may be sufficiently long that the user will worry that nothing is happening unless they are assured otherwise or presented with some method of gauging progress" (ibid. p.90).

This example shows that while the provision of a simple abstraction or façade can help improve the usability of Grid technologies, there is a need for users to be able to query what lies beyond the immediately visible surface.

Grid Scheduling

yesterday/today/tmrw – include TG environment stuff

Users don't always know what they want

Reporting of node load on queued systems

Measuring available memory

Measuring available disk

Wolski- last value is often the worst guess

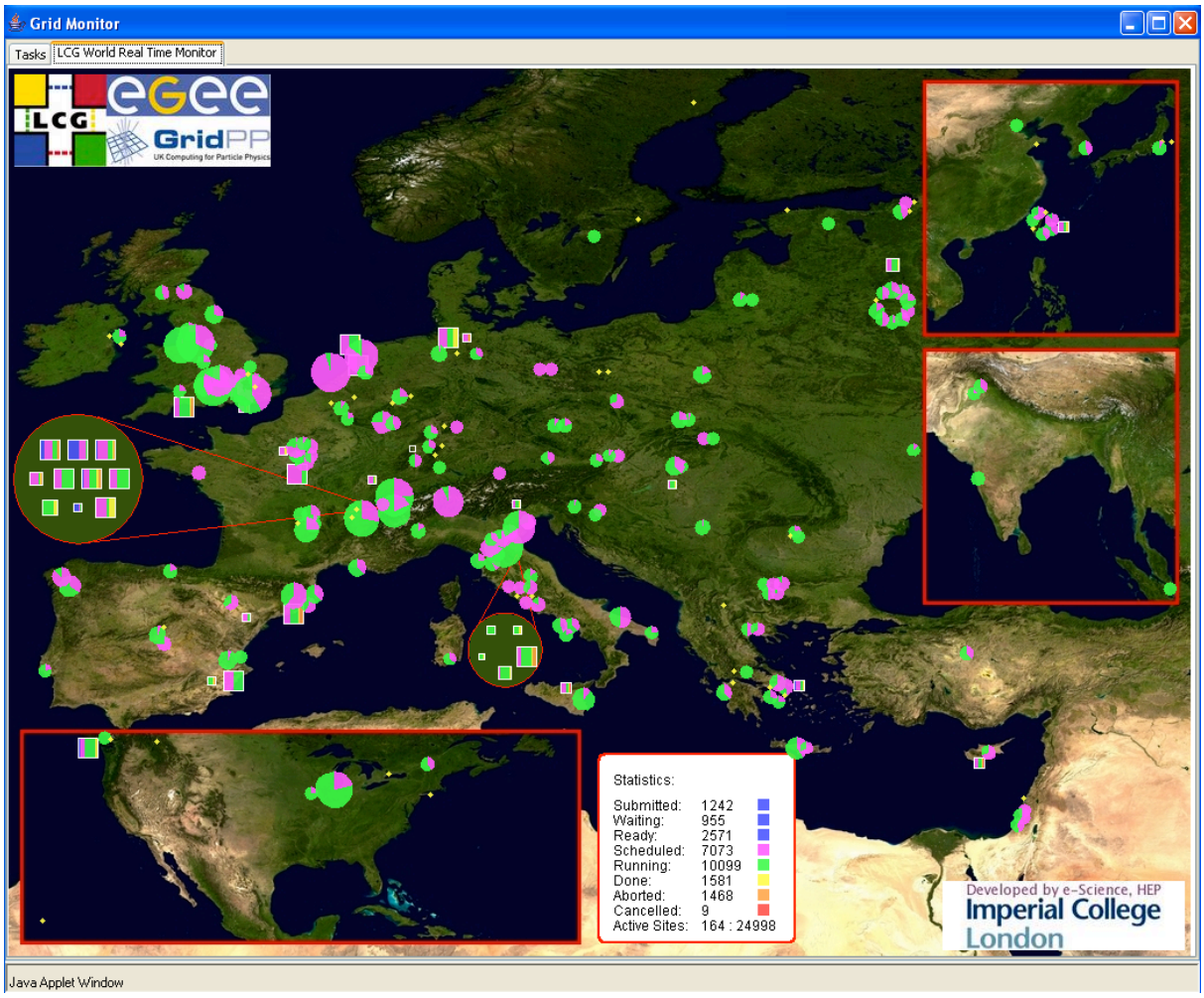

Figure 4: An example of a Grid monitoring tool.

Account Resources and Accountability: Discussion

We see from the preceding discussion that while abstractions are useful tools for protecting users from unnecessary details of system behaviour, rigid adherence to a single abstraction may deny users the information they need to act effectively. As Dourish writes:

"In just the same way as they approach all other activities, they (users) need to be able to decide what to do in order to get things done. In everyday interaction ... accountability is the key feature that enables them to do this. The way that activities are organised makes their nature available to others; they can be seen and inspected, observed and reported. But this feature … is exactly what is hidden by software abstractions ... information about how the system is doing what it does, how the perceived action is organised." (Dourish, 2001: 83)

The question is how do users construct, achieve or make sense of the behaviour of complex infrastructure and its users, and how can we design accountability into it? According to Dourish (2001):

"First we need to find a way to ensure that the account that is offered of the system's behaviour – a representation of that behaviour – is strongly connected to the behaviour that it describes ... Second, we need to find a way to allow this representation to be tied to the action in such a way that the account emerges along with the action rather than separately from it ... Third, we need to ensure that the account that is offered is an account of the current specific behaviour of the system." (Dourish, 2001:85)

To answer this question in the context of Grids, we must take into consideration a number of issues. First, account resources of and for Grid behaviour must satisfy the needs of a wide variety of different users. These differences reflect different user roles (e.g., operational support staff and infrastructure administrators) and different end-user communities. Second, accounts for these users may lie anywhere within what we might refer to as a 'spectrum of virtuality' (see Figure 3) which extends from complete observability (and complete user control) of behaviour at one extreme to complete transparency of behaviour (and lack of user control) at the other. Third, what serves as an adequate account resource of and for Grid behaviour will vary with time, both in the real-time context of interaction and over time as users become more experienced and expert.

The combination of these factors means, of course, that any attempt to identify a priori a finite set of accounts must fail. As we saw with the firewall example, the 'designed for' layered account resource has a finite depth and extent; if, when the user has reached the last account resource, the explanation is still not adequate, the user is still unable to make a decision. Of course, it is relatively easy (technically) to supply account resources with increasing depth, but it is more difficult to increase the extent (or 'breadth') of the account resource, that is, to relate what the application might be trying to achieve, the implications of this in the context of the user's activity, location, and so on, until users have sufficient resources to assemble an account that is relevant to their needs at that moment. Instead, we must look to ways in which users can define and construct their own accounts.

A possible solution is to consider providing flexible tools that will enable users to mine relevant data and visualise it in easy-to-understand ways.

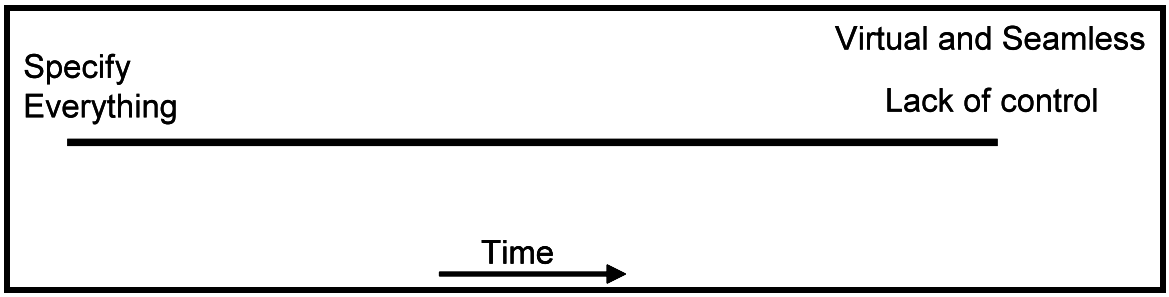

Figure 3: The 'spectrum of virtuality'.

Discussion and Conclusions

When we talk about accountability we do not intend to suggest that Grids can produce accounts in the ways that society members do – they can only provide the resources for the work of assembling an account that has to be done by people. When we say 'provide the resources' for an account we mean to suggest that the modelling of, for example, the system state, the network connection or who has been downloading files from the server are 'docile' – i.e. they cannot be queried in the manner that one might query a human. The docility of accounts produced by systems is one reason for our suggestion that they not be seen as accounts in themselves (if they were they would be rather impoverished accounts) but materials that are constitutive of accounts when assembled, queried, and so forth. As with Garfinkel's clinical records, they do not speak for themselves but have to be made to speak (as it were) by their users – hence our suggestion that they are resources as opposed to fully-fledged accounts. We might want to think of the difference between accounts as produced by systems and those from humans as, to use Button et al's (ref) excellent metaphor, the difference between plastic and real flowers. While the plastic flowers may have some of the same aesthetic appeal of the real, they are deficient in a number of ways – they are imitations that may be good enough for the front desk in the office, but we should not expect them to exhibit the features of regular flowers (odour, for example). This relates also to the ambitions of artificial intelligence – the ambition to produce 'expert systems' and the like is, we would argue, akin to the production of plastic flowers – they may look the part and perform some of the functions of the real expert, but they do not always pass muster in the manner that an expert might be expected to do. It is this essential fragility, born of the fact that these are not fully-fledged experts, flowers or in our case accounts that is important for us. There may be a for our purposes adequacy to an account produced by a system, but to enquire further exposes the essential fragility of the account.

To return to Dourish's comments, above, we would argue that his notion of accounts offered by systems are (mere) representations of their behaviour – they are by no means all that could be said or that we might want to have said. Dourish rightly points to the reflexive tie between accounts and actions described by those accounts – again this points to the use of accounts as a resource (in our case for the production of human accounts). Finally, we should also attend to what it is that the accounts are accounts of – what an account describes may be some high level process that does not speak to our concerns or the obverse (i.e. too low a level to be of any utility). This directs our attention to the central point – accounts are about furnishing the resources to allow one to engage in doing something or other (e.g. seeing that a router needs repairing, spotting an intrusion or logging who has used our data). In short, it is the 'aboutness' of accounts that makes them useful.

What an account is about is fundamental to its utility – this is a commonplace – but how it is about a particular thing in the world is also central, and again it is here that, to our minds, the

systems available seem (albeit inevitably) to fall down. The visibility arrangements of system behaviour are often impoverished and can be seen as generative of their own problem space. Envisioning both operation of and collaboration in e-science systems will remain problematic unless and until (although we are sceptical as to the second part) a more usable system for accounting for system behaviour can be assembled. There will, in short, inevitably be the need for some repair.

While we don't dispute that we must discover ways of enabling use of the Grid to become progressively more routine, we do not equate this with the Grid disappearing behind some high level abstraction. The behaviour of systems must be accountable and we need to understand how to achieve this for different kinds of user.

References

Adams, A. and Sasse, M. A. (1999). Users Are Not The Enemy: Why users compromise computer security mechanisms and how to take remedial measures. *Communications of the ACM* 42(12), pp. 40-46.

Anderson, S., Hartswood, M., Procter, R., Rouncefield, M., Slack, R., Soutter, J. and Voss, A. (2003). Making Autonomic Computing Systems Accountable: The Problem of Human-Computer Interaction. In Proc 1st Int Workshop on Autonomic Computing Systems, 14th International Conference on Database and Expert Systems Applications, Prague, September.

Arsenault, A. and Ferrell, S. (2001). *Securely Available Credentials – Requirements*, IETF RFC 3157.

Balfanz, D., Durfee, G., Grinter, R. E. and Smetters, D.K. (2004). In Search of Usable Security: Five Lessons from the Field. *IEEE Security & Privacy* 2(5), pp. 19-24.

Basney, J., Humphrey, M. and Welch, V. (2005). The MyProxy online credential repository. *Software – Practice and Experience* 35, pp. 801-816.

Beckles, B., Welch, V. and Basney, J. (2005). Mechanisms for increasing the usability of grid security. *International Journal of Human-Computer Studies* 63, pp. 74-101.

Bowers, J. (1994). The Work of Making the Network Work: Studying CSCW in Action. In Proc. ACM CSCW conference.

Dourish, P. (2001). Where the Action Is: The Foundations of Embodied Interaction. MIT Press.

de Paula, R., Ding, X., Dourish, P., Nies, K., Pillet, B., Redmiles, D., Ren, J., Rode, J. and Silva Filho, R. (2005). In the Eye of the Beholder: A Visualization-baseed Approach to System Security. Int. J. Human-Computer Studies.

Ganek, A. and Corbi, T. (2003). The dawning of the autonomic computing era. IBM Systems Journal. 43(1); pp. 5-18.

Gustafson, D., Just, M. and Nystrom, M. (2004). Securely Available Credentials (SACRED) - Credential Server Framework. IETF RFC 3760.

Gutmann, P. (2003). Plug-and-Play PKI: A PKI your Mother can Use. 12th USENIX Security Symposium, Washington. http://www.cs.auckland.ac.nz/~pgut001/pubs/usenix03.pdf

Hardian, B., Indulska, J. and Henricksen, K. (2006). Balancing Autonomy and User Control in Context-Aware Systems – a Survey. Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06), pp. 51-56

Jirotka, M., Procter, R., Hartswood, M., Slack, R., Coopmans, C., Hinds, C. and Voss, A. (2005). Collaboration and Trust in Healthcare Innovation: the eDiaMoND Case Study. Journal of Computer-Supported Cooperative Work, 14(4), p. 369-389.

Kesselman, C. and Foster, I. (1998). The Grid: Blueprint for a New Computing Infrastructure.

Knorr-Cetina, K. (1999). Epistemic Cultures. How the Sciences Make Knowledge. Harvard University Press.

Rogers, Y. (1992). Ghosts in the Network: Distributed Troubleshooting in a Shared Working Environment. In Proc. ACM CSCW conference, November.

Stanyer, D. and Procter, R. (1999). Improving Web Usability with the Link Lens. In Mendelzon, A. et al. (Eds.), Journal of Computer Networks and ISDN Systems, Vol. 31, Proceedings of the Eighth International WWW Conference, Toronto, May. Elsevier, pp. 455-66.