Collaborative construction of Grid: Usability within Particle Physics

Will Venters, Yingqin Zheng, Tony Cornford.

London School of Economics, Houghton Street, London, WC2A 2AE.

w.venters@lse.ac.uk

Abstract. We report the development and use of a Grid (the LHC Computing Grid) to analyse data produced by the Large Hadron Collider (LHC) due to start at CERN in 2008. This study explores the relationship between the parties involved and in particular the role of users (and their virtual organisations) in negotiating the usability of the Grid. Using a case study of a Grid job submission tool called Crab we draw on Andrew Pickering's "Mangle of Practice" to explore how making Grids usable is a practice of experimental physicists as well as a practice of systems designers. The conclusions highlight the need to consider the users of Grids as powerful agents in the development process, able to direct and steer the development to represent their practice.

Introduction

This paper explores the relationship between 'users' and 'systems administrators' during the development of a Grid. Through a study of the development and use of software to support a Grid, we aim to unpick and question the role of user, and the aspiration to make Grids 'usable'. Our work considers how the working practices of scientists shape Grids to reflect their science practice. We discuss how users, as members of experiment collaborations at the Large Hadron Collider (LHC) at CERN, impose themselves on the requirements for Grids. We focus in particular on the negotiation process inherent in the development of any information infrastructure.

The paper presents research undertaken into LCG's (the LHC Computing Grid) development of a computing grid for particle physics. LCG is a worldwide collaboration of particle physicists that aims to build and maintain a data storage and analysis infrastructure for the particle physics community. This is in preparation for the challenge of the imminent "data deluge"(Hey and Trefethen 2002) from the launch of the LHC at CERN, the European Laboratory for Particle Physics in 2008 (previously planned in 2007). The LHC is designed to search for the Higgs boson, a particle predicted by the standard model in physics. Searching for Higgs evidence among the data has been described as like searching one person in a thousand world populations, or for a needle in twenty million haystacks (Britton, Clark et al. 2004). Thus the LHC envisages producing some 15 million gigabytes a year and to process this data LCG envisage requiring 100,000 computers forming its Grid by 2007 and spread across the globe (Faulkner, Lowe et al. 2006). We focus in particular on one of the LHC's four experiments (CMS) and its development of a software suite to enable its experimental physicists to easily run jobs on the LCG. The CMS collaboration consists of 2300 scientists and engineers from 159 institutes in 36 countries developing and running the CMS experiment on the LHC' s 26km ring.

Through an interpretive case study we describe how CMS develop software embodying the needs of users. Data collection has been undertaken through over forty interviews, participant observation in weekly project-management-board meetings of those involved in developing LCG in the UK (a group called GridPP), attendance at major GridPP meetings, site visits and through two week-long visits to CERN. The case study we present is drawn from a number these interviews, drawing in particular on quotes from the individuals in Table I, and various documentary evidence.

Person Code	**Job title**	**Date**
P1	CMS Representative to UK LCG development (GridPP)	Jan 2007
P2	Post-Doc Research Assistant undertaking physics on CMS	March 2007
P3	Post-Doc Research Assistant undertaking physics on CMS	March 2007
P4	Oversees Grid middleware development for LCG and systems administration.	March 2007
P5	PhD student undertaking physics on CMS	March 2007
P6	Involved in middleware development and systems administration based at CERN.	March 2007
P7	A previous senior manager of the integration of experimental software with Grid services.	Jan 2007

Table I.

The analysis of the data collected is based on the premise that we need to examine the collaborative practices of grid development, deployment and use in order to better understand what "usability" of e-science means. e-science has been defined as the "*intersection* of Grid and collaborative research" (David 2004 original emphasis), and Grid technology is often presented as a new generation of distributed computing that can support collaborative distributed activities. Such collaborative activity is undertaken within "virtual organisations using grids" (Foster, Kesselman et al. 2001) of which CMS is an example. Research has been conducted to explore the organisation of scientific collaborations (Chompalov, Genuth et al. 2002) and how such collaborations imposes requirements on the development of Grids as embedded in e-science and e-social science initiatives, such as institutional infrastructures (David and Spence 2003; David 2004), epistemic domains (Fry and Thelwall 2006), and issues concerning methods and theories (Scott and Venters 2007). Even though Grid technology promises great advantages in supporting scientific and business collaborations, its construction, development and adoption cannot be taken for granted, as they are largely collaborative efforts themselves involving substantial community efforts to embed Grid technology into users work practices (Berman, Fox et al. 2003). Examining the collaborative processes of Grid construction, implementation and use can shed light on how the technology can be better integrated into the working practices of scientists in general (Venters and Cornford 2006).

This paper argues that "usability" of technology concerns the integration of a technology into work practices – a process which both socially shapes the technology and the work practices. Given that e-science aims to be disruptive in nature (and by design)– challenging and changing existing science practice (Hey and Trefethen 2002), so a focus on usability in terms of stability and instability of practice is appropriate.

Our wider research explores the active participation of various social configurations in the collaborative construction of LCG. This research responds to calls for a greater understanding of the development and use of e-science technology; "little is known about how, why and by whom these new technologies are being adopted or will be taken up" (Woolgar and Coopmans 2006).

We analyse the case study through a focus on the practice of particle physics and Grid development in the sense of practice as " a routinized way in which bodies are moved, objects are handled, subjects and treated, things are described and the world is understood" (Reckwitz 2002). We draw in particular upon Pickering's (1993) "mangle of practice" in order to explore the relationships of those involved in the development of "usability tools" within the LCG initiative, and the influence of particle physics practice as a dialectic in which material agency is mangled in practice, that is "emergently transformed and delineated in the dialectic of resistance and accommodation" (Pickering 1993). We take the particle physicists' analysis of data from the CMS experiment on the LHC as our unit of analysis, showing how the LCG grid system (both social and technical) performs job submission for CMS, and the role CMS's material media plays in this process for both physicist and systems administrator.

The paper is organised as follows, firstly we describe the practices of CMS physicists submitting jobs to the Grid, outlining in particular the role of CMS' software in this task. Second we describe the impact of CMS software on physicists understanding of failure, and on systems administrators enrolled in this task. Finally we critically assess this case study in relation to usability of Grids for practice through the mangle of practice.

The particle physicist engaging in Grid practices : CRAB.

From a particle physicist's perspective the aim of using the LCG is to undertake analysis which is too large to execute on a local computer or cluster. While "95% of [a researcher's] time is actually spent developing algorithms, developing software, doing analysis, just completely locally on the laptop" (P1) they rely on a robust computing infrastructure to run their large occasional analysis. For them the Grid is "just a processing machine that sits there and nobody really has to know how it works except a few individuals" (P1). They demand the Grid only to undertake analysis as quickly as possible and extremely reliably despite the lack of efficiency of their own code: "People want enough disk space, they want fast CPUs that they can run their usually very inefficient programmes on, and they want it twenty four seven, and also during Christmas" (P2). This high demand for services, coupled by a highly pragmatic approach is characteristic of the field (Hlistova 2004; Venters and Cornford 2006); they are "very dirty programmers, they are not computer scientists and they really will use the fastest way to get at something… they usually want the fast hack" (P2). Indeed to use the LCG physicists must "see it as the fastest way to get somewhere. Because otherwise they will just do it another way. Physicists are not dependent on the Grid, they will choose whatever way, even if that involved people flying to CERN, taking a stack of writable CDs and flying back to London, they will do that. They really do not care. Only if you have a Grid at a workable level is it an alternative" (P2).

In undertaking analysis of the huge quantity of data produced by the CMS experiment individual particle physicists must interact with the Grid. The CMS collaboration has produced a range of software to help its users do this. As stated in this software's manual "The job of the CMS software (online and offline) is to select and process detected events, deliver the processed results to experimenters within the CMS collaboration, and provide tools for them to analyze the processed information in order to produce physics results" (Heavey, Lassila-Perini et al. 2006). Each experiment has produced its own interface to the Grid despite the face that "those interfaces could have been common between the experiments, you could have done it once, and you provide an interface… like a web browser" (P7) but at present this has not been achieved.

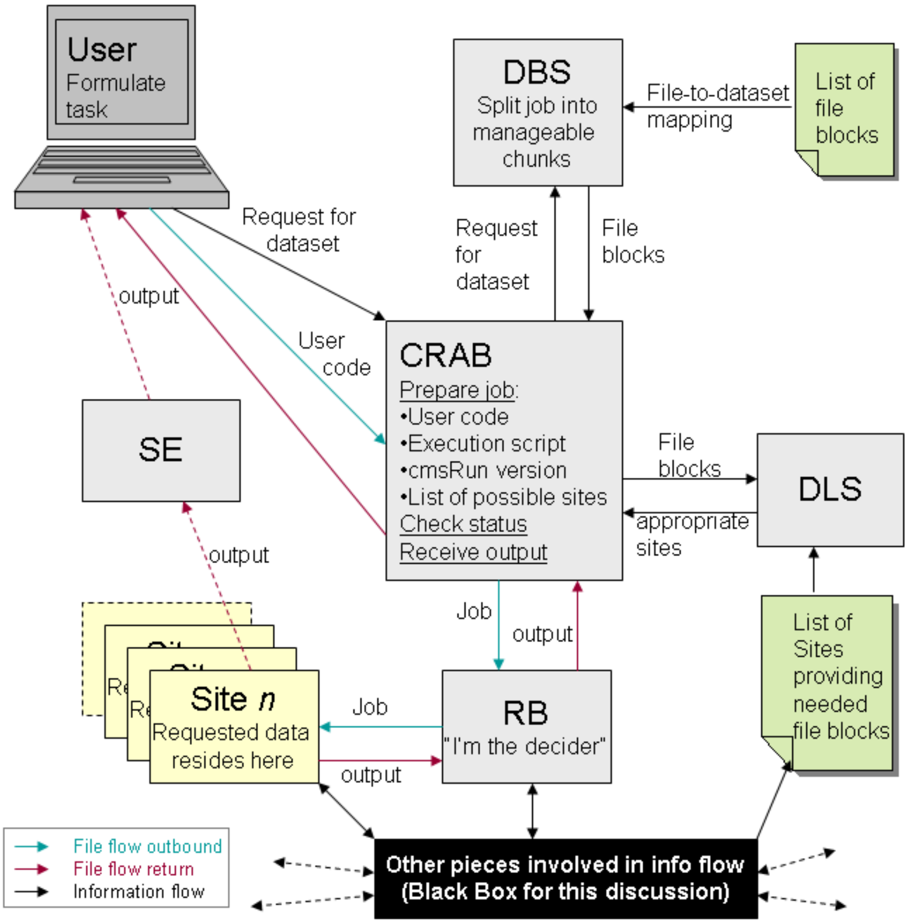

Figure 1 : Executing a job for CMS : (From Heavey, Lassila-Perini et al. 2006).

Interaction with CMS data through the LCG Grid involves a range of stages; writing analysis software to perform the physics analysis (in the C++ programming language), preparing a script (a text file which instructs software) to undertake the analysis "job", executing this script and hence running the job on the Grid, checking the status of the job, then finally receiving the output. Within CMS this execution script is written for software called CRAB (CMS Remote Analysis Builder) produced by programmers within the CMS collaboration. Crab is CMS specific and is "a python program intended to simplify the process of creation and submission of CMS analysis jobs into a grid environment." (Heavey, Lassila-Perini et al. 2006). A user's script is written as a `crab.cfg` configuration file specifying the details of the job. Figure 1, taken from the CMSOffline analysis Workbook, outlines in detail of how Crab then interacts with the Resource Broker (RB), the Data Location Service (DLS), and the Dataset Bookkeeping Service (DBS) to prepare a job for execution on sites, and to move the results to a storage element (SE). By running jobs in parallel this process's speed can be drastically improved. This involves deciding how to break the job into "chunks" and outlining in `crab.cfg` where to get the code and the data, and how to split the job. It must be noted that only Grid sites with CMS software, data and catalogue can run Crab jobs and Crab checks this with the sites everytime it prepares a job. Once a Crab configuration file has been prepared the physicist will run the jobs. The first task is to create a proxy certificate for the jobs they will run (using command "`voms-proxy-init –voms cms`" to create a proxy for the CMS virtual organisation). This proxy provides the job with the same permissions held by the physicist so that the analysis code can be executed on machines (and with data) that the CMS Virtual Organisation has permission to access. To begin the analysis the physicist first creates the jobs using the command "`crab –create X`" where X is the number of jobs, then the command "`crab –submit –continue`" to submit all the created jobs to the Grid to be run. Once running the user can check the status of the jobs using the command "`crab –status –continue`", and finally once all jobs have been completed the results can be retrieved from the Storage Element using the command "`crab –getoutput`".

Physicists usually gain the knowledge of how to produce CRAB configuration files and submit jobs to the Grid through word of mouth: "the way people learn how to use the Grid is by getting a working script from someone… and starting from there and trying to run it and trying to modify is to suite their needs" (P2). In modifying such a script individuals will use websites of FAQs, the CMS documentation, and various Wiki (e.g. https://twiki.cern.ch/twiki/bin/view/Main/CRAB) and Blogs to resolve problems (though seldom to learn the initial steps). They will often e-mail each other scripts for help with problems and for advice.

Breakdown of Jobs on the Grid

Given that the LCG Grid is under development it is inevitable that things will go wrong with Grid jobs, particularly when users regularly run millions of them. We now consider what occurs when jobs fail from the perspective of users, systems administrators, and the role of informational flows such as `crab.cfg` within this process. This allows us to describe the mediating role of CRAB in the process of usability for the CMS collaboration.

Once a job is running the "`crab –status`" command will allow the user to observe the status of their jobs. Users receive a list of their jobs with a status (`ready ¦ scheduled ¦ running ¦ done ¦ clear ¦ aborted ¦ killed`). Once jobs are complete the physicist can retrieve the output from a storage element specified in `crab.cfg`.

One of the complaints made by users is the difficulty knowing what has gone wrong with their jobs. While physicists define success as getting results back from the analysis, those developing grid middleware consider a Grid success as the condition where the Grid has run something, even if the job has crashed or similar problems have occurred. The Grid will return a 0 (success) code in such cases. One interviewee recounted "job-success, was always Zero, i.e. successful, no matter what happened. As long as the grid middleware had submitted a job, that it had run, and had crashed somewhere, they gathered something back, that was considered a success. And this is just not very useful for physicists. For one thing it is impossible to monitor if something is going wrong" (P2). The return of a Zero code represented a passing of responsibility for failure back to the user, rather than with the Grid middleware or systems administrators.

Users were left responsible for failed jobs, and unable to understand the underlying problem, often perceived such failures as a problem with the interface software (Crab) rather than with the underlying Grid: "when things aren't being submitted they say the program is crap, even though it is not the actual program where things are going wrong, it is usually one of the different grid components, not the programme" (Interviewee responsible for documenting Grids). Indeed another interviewee who uses Crab stated "there is a school of thought that the last letter should be changed to P" (P5)!

Resolving such difficulties demands detailed technical expertise, often only held by the systems administrators, and therefore physicists must interact with systems administrators to resolve them or gain a detailed understanding of the Grid themselves. But those interviewed seemed extremely unclear as to the place or person they should approach for help. Users appear to lack information about resources for gaining support. Despite the existence of a Grid support management systems (called GGUS: Global Grid User Support) few users used this, and only if problems occur consistently. Usually users just kill (`crab -kill N -c` where N is job number) jobs which take much longer than the rest and try again with these jobs. CMS's provision of software (such as crab) running on global Grid services creates an artificial barrier to the support. A CMS physicist described: "I know that lots of money is going into support, but it seems to be decoupled, so there is grid support and kind of CMS software support, and I send a software job to the grid and it goes wrong, invariably the problem falls in the gap between the two. So the grid people don't know what I am talking about because I am talking CMS language and the CMS software people say it is a grid problem. So there seems to be this kind of gap at the moment between, almost between the grid and the experiment, that anything falls into that doesn't work, and so on" (P3).

Historically physicists approached problems through informal communication with systems administrators, based on a cultural heritage where systems administrators were local and well known. One interviewee highlighted this "one problems that I have is that when I am…doing computing here (CERN) and it doesn't work, there is someone I can go too to say – 'it hasn't worked, why hasn't it worked?' Whereas in the Grid, sometimes my job ends up in somewhere in Germany, and quite often it doesn't work, and I don't know who to contact, because I am a step removed from the people who are looking after the computers and the jobs that are running".

Without an effective communication route with the systems administrators of the problematic parts of the Grid, users are led to improvise new solutions to these problems. Significantly individual users' needs are represented by their experiments (e.g. CMS) which have significant resources and technical capabilities to respond on users' behalves. CMS is therefore capable of improvising solutions which represent their physicist's needs, whether or not they reflect the needs of the Grid administrators or other experiments (against which they compete for resources). In the words of one interviewee "I really do think that being the systems administrator for particle physicists must be hell. They will all want different systems for everything. And they are all quite savvy, they will tune everything" (P2).

Black-list and White-list.

When problems occur with a site it is the role of the workload-manager middleware software to ensure that jobs submitted to the Grid continue to run (albeit more slowly and less efficiently). "We have a workload manager which is able to do resubmission automatically on behalf of the user in case the site where the job was sent is not behaving correctly. Or we have the possibility in the information system for the VO [Virtual Organisation] to ban explicitly, given sites, so there are white lists and black lists. So [a central Grid administrator] can black-list a site, v can white list another site, if [they] want. So that basically the workload management system is forced to consider all and only the sites left that the application wants to be used"(P4). This feature is only however intended to allow the tailoring of the shared workload manager by administrators.

The aim of the CMS collaboration's software is to "schedule jobs onto resources according to the policy and priorities of CMS" (Heavey, Lassila-Perini et al. 2006) rather than the priorities of the Grid as a whole, or the priorities of other LHC experiments. The CMS response to the problem of difficult to identify failures, poor user support, and problematic areas of the Grid which drop jobs for unknown reasons was to exploit the above middleware facility in order to provide Crab with the facility to force Grid jobs to execute on particular areas of the Grid, or to exclude a particular area of the Grid. Crab therefore now includes the ability to black-list elements of the Grid (so disallowing jobs to run on a site's computing or storage elements (CE – computing elements or SE – storage elements). Similarly Crabs white-listing allowing users to specifically target Grid nodes for a job to run on (see Figure 2). A physicist interviewee recounted that using this facility meant dealing with a problematic return is easy; "in that case I quite often try and send the job somewhere else, not use that particular Grid site" (P2). Similarly another physicist stated: "I use a tool called Crab, which is a CMS tool which allows you to sort of specify places. You can specify places for it not to go to or places that it should go to. So I can try sending it somewhere else, but sometimes what I end up doing is ignoring, not using the Grid at all, and go back to old fashioned, submit directly to a computer here" (P3).

```
If you want/need to select/deselect some site, you can use: (see
Crab FAQ for more info)
  Ce_black_list – (refuse access to all the listed CEs, allow all
others)
  Ce_white_list – (allow access only to those CEs listed)
  Se_black_list – (remove the selected SE from the list of sites
hosting data)
  Se_white_list – (select only the SEs listed)
---------------------------------------------------------------------------------------------
## CE Black List: all the CE whose name contains the following
strings (comma
## separated list) will not be considered for submission.
## Use the dns domain (eg fnal, cern, ifae, fzk, cnaf, lnl,....)
CE_black_list = ...
---------------------------------------------------------------------------------------------
So, in summary, if you want to force your jobs to go a specific
site (eg if you want to test the site), use "SE_w/b_list". If
instead you want to access some dataset but you want to avoid a
site (because you don't trust it), use "CE_w/b_list". In addition,
se_w/b_list cannot be used with None as input dataset.
```

Figure 2: FAQ details of CRAB.cfg options (truncated)

These black-listing and white-listing options appear to break the fundamental aim of Grids in providing "coordinated resource sharing" (Foster and Kesselman 1998; Foster and Kesselman 2004) since they enable users rather than Grid workload management software to coordinate the resources exploited. Indeed the feature was only intended for testing sites, or avoiding specific untrustworthy sites (see Figure 2 final part) yet it appears to now be regularly exploited by CMS users to speed up their analysis. "In practice the way I operate is; I find out where my data is and tell this Crab tool where to go. My experience with telling it just to go and find the data [without using the black-white listing options] is that that does not work, ever. So that may be something specific to do with how far we have got with these tools. But up until now, definitely, from my experience, the kind of real physics applications that I want to run, the only way it works is you find out where the data is and then you tell the thing to send there" (P3).

An interviewee who was developing the Grid middleware explained the problem of experiments producing software for their users in detail: "As people who are trying to put together a middleware release we are trying to find the best solution to each particular problem domain, integrate it into our release so everyone can use it. Now if there was a user sitting in isolation they would probably have to use what we provide, they don't really have much choice. But these users don't really exist so much, they all work for experiments with lots of influence and resources and everything. And which occasionally, possibly often, have very high influence in some of the sites as well, they can ask the sites to install various services. So they can bypass stuff. Bypassing is probably a pejorative phrase, it is just they choose to use an alternative route. (…) But certainly one of [systems administrators] big services is workload management. So the idea is that this takes all your jobs and manages them for you, submits them to the right place, so you send them there and forget about them until you all come back. But on your user interface you can implement most of this stuff, if you want to, to your own satisfaction. And we find people have done that." (P6)

The CMS physicists realize that their approach would create headaches for systems administrators, however their individual focus on physics analysis and the pressure to perform this quickly is paramount in there minds **"**I see that the way we use it may cause other people headaches. But it is simply because otherwise you get into this really boring alternative which is you submit it without specifying where to go. And then it comes back from one place and says it doesn't work. So you say – OK, ignore that place, try somewhere else. You know you are gonna make it work, but you shortcut that by finding out… so this one thing of knowing where you data is, if you can find that out independently it releases more time to make it happen" (P3).

From an LCG system's administrators perspective however the white-listing and black-listing creates significant problems. Within the running of LCG their role is to observe the usage of the Grid through a variety of monitoring and dashboard applications and ensure that the computing elements and storage elements their site contributes to the Grid are running effectively and have the correct software infrastructure installed. This is no easy task, particularly as at the moment the number of storage elements and computing elements are drastically increasing meaning they are often installing new elements. Further the Grid middleware is changing regularly and the powerful experiments regularly demand new software installations. Finally their Grid is likely to be shared by a range of experiments at the LHC, and indeed in other sciences. Such complexity means that the skills of systems administrators are crucial to a site's success. One middleware developer described this problem: "the biggest problems today is managing such a big infrastructure where there are so many sites and not all the sites are managed with the same level of body of quality. At the same time the software which implements the servers is not mature enough in order to be easy to be managed. So clearly, if the computing element was as easy as a web server to be managed, every single site in the world could be reliable. But unfortunately, the computing element of software is not as reliable as the Apache server so this requires a very experienced people in managing sites, which is not always the case" (P4).

Crucially in their preparation for the data deluge once the LHC begins taking data, sites require regular challenges to their infrastructure by large numbers of physics jobs to ensure the site can handle the required volumes. Until the LHC begins operating there is a lack of data to analyse, and therefore a lack of demand for computing resources. There is thus a resultant competition between Grid sites to get jobs to stress their systems, but with physicists able to run their jobs at the sites easiest for them using black and white lists. These are usually sites they know are working well (and hence often do not need as much stress testing) or sites local to them where they can easily contact systems administrators for support rather than through GGUS. In particular CERN has a large number of CEs and SEs which many rely upon. Within CMS users exploit the crab white-list and black-list options to target such sites.

Discussion

The case study shows that for CMS physicists their practices with the Grid have emerged out of a historical culture of resistance and accommodation through improvisation with a focus on physics goals (Zheng, Venters et al. 2007). They demand speed of analysis above other concerns (though they acknowledge that their lack of programming expertise may cause some inefficiency) and appear somewhat individualistic in their Grid demands, yet collaborative in their sharing of files and knowledge. This is played out in their adoption of Grid technology, which they come to understand as a faster, but less robust, form of computing than their personal Clusters or PCs. They accommodate the Grid into their practices through informal interaction among the community, with existing `crab.cfg` files shared and tailored by new Grid users. Crab itself is produced by this community to represent its members' needs and to reduce their Grid interaction to a simple scripting process. Indeed the `crab.cfg` manual discusses little about the Grid as a material artefact, focusing wholly on the process of its use – the practice to do physics. `crab.cfg` is thus an accommodation, and an artefact with agency for its users, and representing their practices.

And yet `crab.cfg` is bounded within a wider system involving failed Grid nodes, and the CMS collaboration's response to such failures. While systems administrators work hard to try to understand and fix problems with the LCG, for some sites this is beyond their individual skills. In this context it is perhaps unsurprising that CMS, and other experiment collaborations, faced with such resistance to their physics have accommodated (Pickering 1993) it by the ability to black-list specific poorly run sites. Similarly if sites are to be effectively tested for their compliance with CMS analysis the feature of white-listing is required. CMS's approach to the problems with LCG has been to tailor its material resources (in particular the CMS software including Crab) to provide its physicists with agency over failing sites.

By allowing black-listing and white-listing individual CMS physicists were able to respond to their own perception of problems on the Grid (another resistance) by targeting sites they believe work effectively, or by black-listing sites which they believe fail, and so speed up their analysis process on a Grid with plenty of resources available. These pragmatic physicists adopted this practice not as an extreme means of testing or sanction, but as common practice for interacting with the Grid. Such targeting might however ossify over time as the shared "working" `crab.cfg` scripts (which include black-lists or white-lists) are passed on. Only once the white-listed grid sites become overloaded, and hence further resistance to physics practice occurs, might accommodations be explored such as trying black-listed sites again, or changing the white-list by changing `crab.cfg`. This is perhaps unlikely to occur while computing and storage are plentiful, but once the LHC starts to take data problems may well occur (though it is highly likely that the features on which black-listing and white-listing rely would be withdrawn if huge numbers exploited it. Until then however the problem remains).

On the other side of this process lie systems administrators. For them the main agent for allocating jobs to sites should be the Grid middleware (through the workload manager). They argue that were every Grid user to employ black/white-listing then "the impact ultimately … [would be] that the workload management system would be scrapped, because it would be investment in something useless" (P6). Yet their monitoring of the Grid and its success is distorted by the `crab.cfg` targeting of jobs. They must attempt to continue to develop their Grid resources without a balanced view of Grid activity, and with the workload-manager being sidelined by targeted `crab.cfg` scripts. For systems administrators, and middleware developers, are faced with a powerful and technically proficient user community developing applications which exploit the Grid for their own particular needs, rather than accepting it as a homogenous service to all. While Crab is an example of this, it is not the only one. “The experiments try to do even more than what we provide as let’s say, common software. So they develop independent monitoring systems that report independently the situation of the infrastructure as they see it. Not as the managers see it. And they develop client code that is able to do as much as possible in the automatic way of clean-up” (P4). Those developing the middleware as simply hopeful that as their middleware improves more of the experiments will move to use it rather than continuing to develop their own application software representing their own needs.

Conclusions

We draw on Pickering’s (1993) “mangle of practice” which suggests that the practice of science is based on messy ill-defined material artefacts through a dialectic of resistance and accommodation that shifts CMS physicists through the space of all potential arrangements they can think of (Pickering 1993). Through this we argue that usability of grids is accommodated (and in turn influenced) by the human agency of users. In this case study human agency in doing physics-with-grids were temporally emergent from the practices of doing physics (with its focus on results). Resistance emerged from doing physics (hence Zero return codes were problematic as they failed to focus on doing physics), and accommodations (by CMS) were made within the general practice of physics in which individuals faced with resistance improvise and bricolage solutions which they share with the collaboration. The Grid-in-use is thus not technologically deterministic in its influence on practice, but rather negotiated and contested between material agency of the Grid, and the human agency of users as they accommodate its various resistances in their purposeful pursuit of physics through the mangle of practice (Pickering 1993).

What then for the “designer” of Grids? Developing a Grid for e-science is perhaps to undertake an exercise in co-aligning technical and social elements within the practice of users. Our study suggests, that Grid ‘users’ in particle physics are not passive in this realignment process, but rather, as a virtual organisation (CMS), were active agents responding to the grid as a part of their mangle of practice. We respond to the assertion of the need to challenge the designer/user opposition (Suchman 1999) by recasting the roles of users and developers in LCG (a theme espoused by Woolgar & Coopmans (2006)). We suggest that for LCG the ‘user’s’ practice involves responding to resistance in usability by accommodations which may themselves be technical Grid developments (like Crab). The process of design of Grids, in this case, is a co-evolution of the accommodations of the designer (attempting to produce a generic Grid middleware for all) and user (attempting to produce an accommodation which allows their physics goals to be achieved). Indeed here the desire to create standardised Grid interfaces (analogous to the electrical power Grid) is proved problematic by users attempt to create specialist interfaces tailored to the specific needs of their practice.

Acknowledgement

This research was undertaken as part of the Pegasus project (Particle-physics Engagement with the Grid:A Socio-technical Usability Study) funded by the UK EPSRC (Grant no EP/D049954/1). Further details available at: www.pegasus.lse.ac.uk

References

Berman, F., G. Fox, et al. (2003). ‘The Grid: past, present, future’. *Grid Computing - Making the Global Infrastructure a Reality*. F. Berman, G. Fox and T. Hey, John Wiley & Sons, Ltd.

Britton, D., P. Clark, et al. (2004). ‘A Grid for Particle Physics - from test-bed to production’, GridPP.

Chompalov, I., J. Genuth, et al. (2002). ‘The Organization of Scientific Collaborations.’ *Research Policy*, vol. 31, pp. 749-767.

David, P. A. (2004). ‘Towards a cyberinfrastructure for enhanced scientific collaboration: providing its 'soft' foundations may be the hardest part’. Oxford, Oxford Internet Institute.

David, P. A. and M. Spence (2003). ‘Towards institutional infrastructures for e-science: the scope of the challenge’. Oxford, Oxford Internet Institute.

Faulkner, P., L. Lowe, et al. (2006). ‘GridPP: development of the UK computing Grid for particle physics.’ *Journal of Physics G: Nuclear and Particle Physics.* vol. 32, pp. N1-N20.

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

Foster, I. and C. Kesselman (2004). *The Grid 2: Blueprint for a New Computing Infrastructure*, Elsevier.

Foster, I., C. Kesselman, et al. (2001). ‘The Anatomy of the Grid: Enabling Scalable Virtual Organizations.’ *The International Journal of High Performance Computing Applications,* vol. 15, no. 3, pp. 200-222.

Fry, J. and M. Thelwall (2006). ‘Using Domain Analysis and Organisational Theory to Understand E- Science Sustainability’, *Second International Conference on e-Social Science*, Manchester, UK.

Heavey, A., K. Lassila-Perini, et al., Eds. (2006). *The CMS Offline Workbook,* CERN.

Hey, T. and A. Trefethen (2002). ‘The UK e-Science Core Programme and the Grid.’ *Future Generation Computer Systems,* vol. 18, pp. 1017-1031.

Hlistova, J. (2004). ‘How is the Particle Physics Community Shaping the Grid in the UK?’ *Department of Information Systems*. London, London School of Economics, no. 38.

Pickering, A. (1993). ‘The mangle of practice: Agency and emergence in the sociology of science.’ *American Journal of Sociology* ,vol. 99, no. 3, pp. 559-589.

Reckwitz, A. (2002). ‘Toward a Theory of Social Practices:A Development in Culturalist Theorizing’. *European Journal of Social Theory,* vol. 5, no.2, pp. 243-263.

Scott, S. V. and W. Venters (2007). ‘The practice of e-science and e-social science: method, theory, and matter’. *Virtuality and Virtualization,* Springer, Boston.

Suchman, L. (1999). ‘Working relations of technology production and use’. *The Social Shaping of Technology*. D. Mackenzie and J. Wajcman. Maidenhead, UK, Open University Press**,** pp. 258-269.

Venters, W. and T. Cornford (2006). ‘Introducing Pegasus: An ethnographic research project studying the use of Grid technologies by the UK particle Physics Community’. *Second International Conference on e-Social Science,* Manchester, UK.

Woolgar, S. and C. Coopmans (2006). ‘Virtual Witnessing in a Virtual Age: A prospectus for Social Studies of E-Science’. *New Infrastructure for Knowledge Production*. C. Hine. Hershey, Information Science Publishing**:** 1-25.

Zheng, Y., W. Venters, et al. (2007). ‘Agility, Improvisation, or Enacted Emergence’. *International Conference on Information Systems*, Montréal, Canada, Association for Information Systems.