

“CLUSTERS OF COMPETENCE” IN THE DISTRIBUTED DEVELOPMENT OF GRIDS: A PARTICLE PHYSICS COMMUNITY’S RESPONSE TO THE DIFFICULTIES OF GLOBAL SYSTEMS DEVELOPMENT.

Authors: Avgousta Kyriakidou and Will Venters
London School of Economics and Political Sciences
Information Systems and Innovation Group, Department of Management

Abstract

This paper examines the distributed development of Grid infrastructure in the particle physics setting. Specifically, the focus of concern is the collaborative practices employed by particle physicists in their attempt to develop a usable Grid with the aim to offer lessons to those involved in globally distributed systems development.

1 INTRODUCTION

Grid computing promises to distribute and share computing resources “on tap” and provide transparent communication and collaboration between virtual groups (Foster and Kesselman 2003). Yet developing and implementing such complex information infrastructures requires collaboration among a range of dispersed groups, and flexibility and adaptability to volatile requirements (Berman, Geoffrey et al. 2003). Here, we examine a case-study of Grid development within particle physics, the LCG (Large hadron collider (LHC) Computing Grid), in an attempt to explore how such a large-scale distributed system is developed collaboratively in a global way in readiness for data from experiments at the recently launched LHC at CERN, Geneva. This year the LHC particle accelerator, will begin to collide protons in a search for the “Higgs boson” particle (Doyle 2005) and so produce 12-15 Petabytes of data annually. The storage and analysis of the data requires a Grid of 10 Petabytes and of 100000 CPUs. Traditionally such resources would be centralized at one location, however, in the case of the LCG, various external pressures (such as funding) mean a novel globally distributed Grid is required.

Grids are a new form of large-scale systems that are highly distributed, both in conception/construction and operation (Foster and Kesselman 2003). Developing a Grid is argued to be a significant systems development challenge because it must be done as a distributed collaborative effort (Berman, Geoffrey *et al.* 2003). The systems development setting studied here is the particle physics community which is well-known for the development of other cutting edge distributed systems to support their work (most notably the web) and is itself highly distributed, so presenting a context where distinctive collaborative practices emerge.

Exploring this case we argue that the development of Grids poses new and underexplored opportunities for understanding collaborative global systems development. We particularly examine the collaborative practices of the particle physics community as it develops its Grid with the aim to offer answers into the wider context of distributed systems development. In studying these, we consider practices as an emergent property linked to improvisation, bricolage and dynamic competences which unfolds as large-scale projects evolve.

The work begins from the assumption that systems development beyond the smallest scale is inherently a collaborative activity (Whitehead 2007). Yet, with the globalization of the development process (Sengupta, Chandra et al. 2006) and new forms of open technologies such as Grids, the limitations of traditional systems development practices (and even contemporary Agile practices) have become obvious (Parnas 2006; Fitzgerald 2000). Global systems development (GSD) therefore demands new, different development practices, since the nature of the problem and the environment are different (Herbsleb and Moitra 2001). Russo and Stolterman (2000) and Fitzgerald (2000) argue that new systems development

practices, or guidelines for successful systems development, should be drawn from “best practice” development situations that prevail today. They further argue, researchers should focus on examining systems development practice in real-world situations, and in real-world development projects. This paper rejects the idea of “best-practice”, instead exploring a case which is interesting and unusual in the way they develop systems and therefore provides a juxtaposition to more orthodox accounts with lessons emerging from reflection on both.

Our theoretical framework is drawn from activity theory, and frames the LCG project as a complex activity system influenced by the context, the community’s rules, norms, culture, history, past-experiences, shared-visions and collaborative practices (Nardi 1996). We understand the Grid’s development as a series of contradictions between the elements of this activity system, which are in a continuous process of getting resolved in order for the activity system to achieve stability and balance. Contradictions are considered to be the major source of dynamism and development in activity theory (Bertelsen 2003), since based on the emerging problems and conflicts, people have to re-considered their position and collectively re-construct their shared understanding, knowledge and practices.

In addition to empirical evidence from field-work with LCG members and at CERN, the research draws upon literature from fields such as global systems development, open-source development, and global-outsourcing to provide concrete practical recommendations for those considering the collaborative development of large-scale systems, and provide suggestions for how such collaboration might be aided. The following section reviews these literatures. Section 3 and 4 present the theoretical framework and methodology. The case study then follows, after which analysis is presented. Finally, tentative conclusions for the information systems development and Grid communities are provided.

2 LITERATURE REVIEW

With the current trend of globalization and the problems of turbulent business environments (Herbsleb, Paulish et al. 2005), the IT industry has turned toward globally distributed software development in an attempt for the silver bullet of high-quality software delivered cheaply and quickly (Agerfalk and Fitzgerald 2006). Ongoing innovations in information and communication technologies (ICTs) have made it possible to cooperate in a distributed fashion. From originally quite small co-located projects, enabled by technological advances, companies now embark on major complex software development projects running in geographically distributed environments (Oshri, Kotlarsky et al. 2008) and GSD is therefore “becoming a norm in the software industry” (Damian and Moitra 2006).

GSD is claimed to be one of the megatrends shaping the industry today (Simons 2006). However, it presents a special challenge not because it introduces new ways for software to fail but because it drastically complicates communication, coordination and control and changes the nature of the development environment (ibid). The changing nature of the environment and the “faster metabolism” of business today, require organizations to act more effectively in shorter time-frames and to develop software at internet speed (Ramesh, Cao et al 2006). Furthermore, the access to a larger pool of expertise, often in low-cost geographical locations (Kotlarsky and Van Fenema 2008), compels companies to form virtual alliances with other organizations in order to survive.

The distribution of companies has become a necessity not only because of the need of shifting labour markets, but also in pursuit of talented people regardless of location (Ye 2005). Particular attention is hence being given to the opportunities and difficulties associated with sharing knowledge and transferring “best practices” within and across organizations (Orlikowski 2002). Knowledge-based collaboration within systems development has therefore become important (ibid). Similarly the global outsourcing literature suggests that distributed collaborative practices in systems development are important and timely (Lacity and Willcocks 2001; Yalaho 2006). Over the years the practices involved in GSD have undergone refinement and new more effective practices that focus on the collaborative factor have emerged such as agile practices/agility (Nerur, Mahapatra et al. 2005).

Unlike traditional development practices, agile approaches deal with unpredictability by relying on people and their creativity rather than on processes (Cockburn and Highsmith 2001) and are adaptable to project specific circumstances, such as the experience of the team, customers' demands etc. (Bajec, Krisper et al. 2004). They are characterized by short iterative cycles of development driven collaborative decision making, extensive communication, rapid feedback, parallel development and release orientation, features which show their ability to respond to change and create innovation (Highsmith 2003).

GSD is considered to be the new paradigm in developing large-scale systems (Damian and Moitra 2006). However, there are still challenges involved in managing the development, such as communication issues and technical issues that need to be addressed (Herbsleb and Mockus 2003). Globalization increases the complexity and uncertainty of collaborative development effort, which can in-turn negatively influence project outcomes (Lee, Delone et al. 2006). While literature stresses for practices and processes that are flexible and adaptable to the increasingly volatile requirements of the business environment (Highsmith and Cockburn 2001), Lee, Delone et al. (2006) argue that successful GSD requires not only flexibility but also rigor in order to cope with complex challenges and requirements of global projects. There is an ongoing debate which rejects the idea of agile practices as a silver bullet to the challenges of GSD (Parnas 2006). Although, agile practices can work perfectly well in small, self-organized co-located teams (Boehm and Turner 2004), there is an urgent need for scaling agility to incorporate distributed and collaborative systems development situations (Zheng, Venters et al. 2007).

GSD is argued to be “a discipline that has grown considerably richer through practice, influencing research and established practices themselves” (Damian and Moitra 2006). However, the practices and methods employed are far from mature and fully understood (Herbsleb and Moitra 2001). With GSD the limitations of traditional systems development practices become even more obvious (Hanseth and Monteiro 1998). Furthermore, there is a fundamental shift from the development of traditional IS to the development of global information infrastructures, such as the Grid (EuropeanCommission 2006). Grid infrastructures should be seen and treated as large-scale and open as they demand collaborative development in a global/distributed environment (ibid), an environment characterized by high uncertainty and complexity and a continuous stream of improvisation, bricolage, drifting, mutual negotiation, regularity, progress and cycles of interactions (Nandhakumar and Avison 1999). Development often requires ad-hoc problem solving skills and creativity, skills which cannot easily be preplanned (Ciborra 2002). GSD for large-scale systems demands new, different systems development practices as the nature of the problem is now different. Long-cherished computer science principles and early systems development are therefore re-examined in the light of the new requirements.

3 THEORETICAL FRAMEWORK

In contrast to the deterministic views inherent in much of the literature on Grids we employ Activity theory (AT) as an approach to help us look at how technology is collaboratively constructed to fulfil the objectives of a global community (Nardi 1996). AT has inspired a number of theoretical reflections on what information systems development (ISD) is about (Kuutti 1991; Bertelsen 2000). It is argued that AT can provide a theoretically founded but detailed and practicable procedure for studying ISD as a real-life collaborative work activity in context (Korpela, Mursu et al. 2002) rather than as an individualistic process of interaction devoid of such social context.

AT provides a well developed framework for analyzing the complex dynamics of collaborative settings which typically involve interacting human and technical elements (Crawford and Hasan 2006). The concept of collectiveness and the notion of different actors sharing the same goals and constructing the same meanings are at the core of this theory (Leontiev 1978), and are vital to our analysis of an “exceptional” community. AT's focus on accumulating factors that affect the subjective interpretations, the purpose, and sense making

of individual and group actions and operations, also provides a useful paradigm for the ways in which human experience, needs, collaborative practice and creativity shape the development and effectiveness of emerging technologies (Crawford and Hasan 2006) and therefore make this theory suitable for this study.

Vygotsky (1978) originally introduced the idea that human beings' interactions with their environment are not direct but rather they are mediated through the use of tools. Inspired by this concept, Engestrom (1987) extended Vygotsky's original framework to incorporate Leontev's social and cultural aspects of human activity, which reflect the collaborative nature of human activity (figure 1).

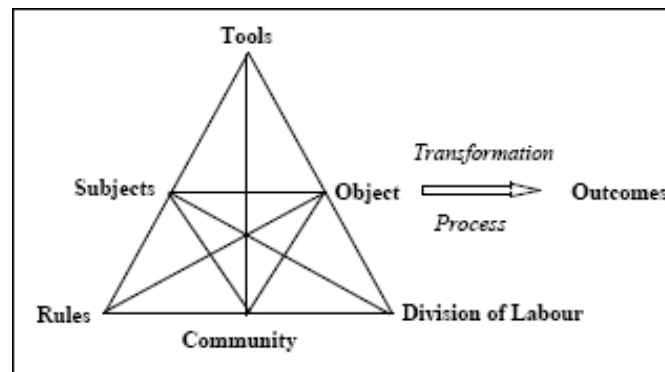


Figure 1.

The fundamental unit of analysis in AT is the entire human activity in the collective context (Guy 2003) The activity itself is the context and is undertaken by human agents (subject) who are motivated toward the solution to a problem (object) and mediated by tools (artifacts, ISD methodologies, practices, cultural means etc.) in collaboration with others (community). An activity is therefore social within a community and influenced by the community. The structure of the activity is constrained by cultural factors including conventions and norms (rules) and social strata (division of labor) within the context (Mwanza 2001).

Activity systems are driven by communal motives that are difficult to articulate for individual participants (ibid). They are in constant movement and internally contradictory. “Their systemic contradictions, manifested in disturbances and mundane innovations, offer possibilities for expansive developmental transformations” (Engestrom 2000). Such transformations proceed through stepwise cycles of expansive learning which begin with actions of questioning the existing standard practice, then proceed to actions of analyzing its contradictions and modelling a vision for its zone of proximal development and then to actions of examining and implementing the new model in practice (ibid). AT with the concept of contradictions, provides a conceptualization of collaborative breakdowns and tensions, which are viewed as highly important in understanding collaborative systems development (De Souza and Redmiles 2003). Contradictions reveal themselves as breakdowns, disturbances, problems, tensions or misfits between elements of an activity or between activities (De Souza and Redmiles 2003) and are considered to be the major source of dynamism, development and learning in AT (Bertelsen 2003)

4 RESEARCH METHODOLOGY

4.1 Research contexts

Grid technology is claimed to be a fundamental step towards the realization of a common *service-oriented infrastructure* for on-demand, distributed, collaborative computing, based on open standards and open software (Foster, Kesselman et al. 2003). It aims to provide a

transparent, seamless and dynamic delivery of computing and data resources when needed, similar to the electricity power Grid (Chetty and Buyya 2002; Smarr 2004), and in this way enable the sharing of computer processing power, storage space and information on a global scale (Berman, Geoffrey et al. 2003). Carr (2005) brashly suggests that the shift to Grid computing forms of technology will “overturn strategic and operating assumptions, alter industrial economics, upset markets and pose daunting challenges for every user and vendor”. Similarly Berman *et al* (2003) suggest that Grid infrastructure “will provide the electronic foundation for a global society in business, government, research, science and entertainment”. While these are obviously extremely bold predictions, and term Grid remains ill-defined, Grids remain an important step towards global IT infrastructures. A Grid is just a large number of distributed processors and other computing devices linked through networks, and presented to the user as a single computer and without the need to address individual resources directly (unlike the web in which URLs address particular web-serving machines). Among many international Grid projects worldwide, particle physics stands out, because of their exceptional distributed collaboration (Chompalov, Genuth et al. 2002), their significant contribution to Grid’s development and the fit of their style of analysis to Grid’s capabilities. Being the first scientific community to be involved in the development of such a large-scale infrastructure, their contribution can be influential in informing the way other communities develop, adopt and conceptualize large-scale systems in general and the Grid in particular.

4.2 Research design

LCG’s collaboration uniqueness, prevents comparative studies, but provides a revelatory case of distributed systems development practice. An interpretative case study is thus used to gain in-depth understanding of the dynamic, complex, loosely-coupled particle physicists’ systems development activity and the collaborative construction of their shared practices. Research evidence was collected through over 70 semi-structured interviews with key members of LCG as well as observation of major meetings/workshops and three week-long trips to CERN. Reviewing of LCG’s documentation was also carried out. Interviews were audio-recorded, transcribed and coded with Atlas.ti, though this was used in a flexible means as a device to aid interpretation rather than for detailed coding.

5 CASE STUDY: THE LCG PROJECT

On September 10th 2008 the Large Hadron Collider at CERN in Geneva began particle acceleration (though quickly stopped due to magnet failures). When fully operational this accelerator will produce six hundred million interactions per second and, after considerable filtering, 15 million Gigabytes of data annually, requiring a unique amount of processing power (100,000 CPUs) and storage space to allow the thousands of physicists globally to analyze them (Lloyd 2006). The proposed solution to this and other computational-intensive and data-centric problems is the Grid (Colling 2002; Lloyd 2006).

The LCG project’s mission is to build and maintain this Grid infrastructure for the entire high energy physics community that will use the LHC (*ibid*). Building the LCG is a highly distributed, complex and poorly defined systems development task. Cutting edge technology and tools are used, new standards reflecting security issues etc. are being negotiated and middleware (which is like the operating system of the Grid) together with other supporting software are being developed collaboratively by physicists around the world. Particle physicists have a long tradition of such large-scale global collaborations and working on a distributed basis is just a part of their everyday routine (Knorr-Cetina 1999). Indeed, building this large-scale, by nature distributed Grid, demands global development. Firstly, funding needs to be taken by different sources and secondly an enormous amount of manpower is needed for the different Grid elements to be developed. These dictate that Grid elements be globally distributed rather than collocated at CERN.

The systems development activity of LCG is organized into a number of projects, some of which extend beyond the physics community. The fact that funding is so difficult to get and it is politics rather than technology which may inhibit the success of such Grid initiatives

(Kyriakidou and Venters 2007), means that other people beyond physics need to be involved, in order to ensure transferability and usability in other disciplines. Other operational Grid organizations providing resources for the LCG are the EGEE (Enabling Grids for e-Science) which is jointly producing middleware with LCG, the OSG (Open science Grid), the GridPP (UK's contribution to LCG) etc. Participants in this large-scale collaboration include by and large particle physicists, however a number of computer scientists, software engineers and people from other advanced sciences are also active members in the development, deployment and user support.

Particle physicists' collaborative work practices are not typical (Knorr-Cetina 1999; Shrum, Genuth et al. 2007) and have been described by Chompalov, Genuth et al.(2002) as "exceptional". LCG's constitution reflects these work practices and is thus based on a collaboration where decisions are made based on democratic and consensual basis with minimal levels of internal authority (Traweek 1988; Shrum, Genuth et al. 2007). Particle physicists are highly interdependent upon each other's work for the generation of scientific results, which creates a trustful environment in which management is minimised. The management structure of the LCG is flat and it is best described as a network rather than a hierarchy.

The systems development activities undertaken by the LCG project varies. They include the development of middleware components, installation and maintenance of Grid hardware, development of physics applications for job submission to run on top of the Grid middleware, testing and certification of applications, ensuring patches have been installed and user support. Despite having recruited some computer scientists for the project, common practices (emerging from particle physics) are that software code is written on an ad hoc basis so as to make things work, rather than following any standard system development practices. Furthermore, because the Grid is a new and emerging technology, LCG cannot take a plan-based approach to development and that's why they are pragmatic with the aim to improve by trial-and-error. The Grid is already partially in use, and thus some physicists are already users who write software to undertake their analysis. Particle physicists are powerful users, since they have computing expertise and therefore tensions exist between them and the specialist developers. Most meetings for coordinating and dealing with issues around the development are conducted virtually. Video-conferences between groups of people responsible for each development task, or exchanging of emails are two standard ways of working. Knowledge is located and socialized through these shared resources as well as through key individuals, who are considered experts and carry out such knowledge and expertise by attending different meetings and by constantly changing job posts.

6 ANALYSIS

The scope of this paper is not to undertake a whole activity theory analysis and therefore provide in depth interpretations of the activity's structure (activity-actions-operations). Rather, we aim to create a rich understanding of the LCG development activity and how this is influenced by the distributed context, the community's rules, norms and collaborative practices. We do this by constructing the developers' activity system and so identifying some of the frictions and tensions through the AT concept of contradictions.

LCG project's activity systems

Identifying an activity system (according to Engestrom's model) requires the identification of the following components: activity of interest, object, subjects, tools, rules, division of labour and community. Findings so far indicate that the LCG project involves two sub-activity systems which we term the developers' activity system and the users' activity system. The focus of this paper is on the developers' activity system.

	Developers	Users
<i>Activity of interest</i>	Distributed collaborative systems	Using the Grid for physics analysis

	development	
Object	Development of the Grid infrastructure for supporting the LHC	Run analysis software on LHC data using the LCG Grid
Desired outcome	The LCG enabling analysis such that the LHC can achieve breakthroughs in physics	Successful analysis of their data, do physics
Subject	Particle physicists, computer scientists/software engineers	Mainly particle physicists
Tools	Systems development practices and methodologies, programming languages	Grid infrastructure
Community	Collaboration, high degree of competence, shared goals, trust, pragmatism etc.	Collaboration, high degree of competence, shared goals, trust, pragmatism etc.
Rules	Collaborative way of working, deliver on time within the budget etc.	Collaborative way of working, finish their PhDs or post-doctoral research.
Division of labour	Fuzzy limit, everyone is doing what does best. People can have more than one jobs. Limited lines of authority. There is no leader or manager, there are spokespersons	Fuzzy limit, everyone is doing what does best. People can have more than one jobs. Limited lines of authority. There is no leader or manager, there are spokespersons

Table 1.

Developers' activity system

Particle physicists are waiting for the LHC (**desired outcome**) to begin to fully operate where upon it will produce large volumes of data for analysis. The objective behind the LCG's systems development activity was realized when particle physicists understood that in order for the LHC to be successful they needed a Grid to undertake analysis (**object**). This realization made particle physicists form global distributed virtual alliances with a number of actors such as funding bodies, universities and the industry, since the development of such a large-scale global infrastructure needed to be done collaboratively as a community effort. As one interviewee stated: *"What physicists want to do cannot be done by a small group, it needs a large collaboration"*. The people involved in the development of this global infrastructure come from different universities and institutes around the globe (including CERN) and involve particle physicists with high technical computing skills, as well as traditional computer scientists/software engineers (**subjects**).

Their activity is driven by the imperative to analyze data from the LHC. As Grid technology is new and different they argue that they cannot take a plan-based approach to systems development. Their aim is to learn and move forward by trial and error. Furthermore, the complexity, pressures and scale of the project mean that no-one can have a full and clear overview of the system and therefore requirements are difficult to pre-specified in detail, the architectures are developed based on assumptions and even the one-centrally designed piece of technology, the EGEE middleware, is modularized and released gradually. The response of LCG to this is to pragmatically and creatively react, drawing on the down-to earth and creative approaches (**tools**) embedded in the particle physics tradition and history.

Particle physicists have themselves recognized their computing practices as amethodical, highly pragmatic and improvisational. The lack of formal processes in systems development is openly acknowledged, with most believing their existing work practices are effective given their primary purpose of building a working system in an extremely limited time-scale: *"[physicists] are more pragmatic in computing"* *"When particle physicists do things, they do them just to resolve a problem for now"*. Physicists follow a bottom up and reactionary approach to development. However, such a distributed development environment requires flexibility and adaptability to changing requirements and external pressures and for this reason they make use of other more flexible practices. Developers have short-term goals and

therefore have short cycles of iteration with continuous releases. Releases are usually tested and certified before they move into the pre-production Grid, where further robustness tests and feedback are gained. Developers also use prototypes for feedback, to improve functionalities and to gather requirements. Experience plays a crucial role as most of the developers are the physicists who themselves know what needs to be done. Concerning traditional systems development methodologies, the usage is little: As an interviewee claimed: *“I would say that we use methodologies but just up to the limit that it is appropriate for what we are trying to do...You will have regression testing and all this sort of stuff in software development and there is a whole life cycle of requirements and specification gathering...but also things change, [so] how can you possibly do a formal software engineering approach to something if we are going to change it ? So software engineering is used up to a point but certainly not completely religiously”*.

A distinct feature of identifying and exploiting technical solutions in the project is the reliance on natural selection. Within particle physics the use of competing technological solutions is a traditional way of working – often as people simply try to solve their problems without consulting others (Pickering 1995). Physicists *“are powerful users, they do what they want. It’s in their culture, it’s their mentality. If they need something, they will just going to get it or they will do it by themselves, although it might already exists”*. Once these hacked solutions exist natural selection selects those to be carried forward, e.g. because of technical failures, lack of funding etc., rather than politics or social power defines the one to be followed: *“The cream comes to the top. Things that work win out and that’s how we worked it”*.

Particle physicists see themselves (and are seen by many) as the “elite among the sciences and as the elite among the physics” (Traweek 1988). Traweek (1988) described them as “promethean heroes of the search of the truth” and outlines the inherently collaborative nature of their community, with Knorr-Cetina (1999) similarly describing them as communitarian (**community**). Collaborative working has traces back in their history, their culture and the nature of their experiments seen as collaborations: *“Particle physicists have collaborated anyway...we run big collaborations across every nation on earth and they always work”*. Their experiments are collaborations requiring collaboration between large numbers of people, usually globally distributed and dissembled when the experiment finishes. For particle physicists, the development of the Grid is like all other previous collaborations they have run, hence like all other experiment collaborations. LCG, to a certain extent, is also set up in the model of experiment: *“The original proposal was set up deliberately to make it look like an experiment. This whole idea of collaboration board for instance, comes out of the idea of what an experiment does”*. People take part in this global distributed collaboration and openly share their knowledge because they like feeling that they have contributed to the collective cause. They are driven by the shared goals and “sacred causes”. As one interviewee put it *“We are trying to get to the data that comes out of the LHC and there are times which we know that we will compromise our own parochial little gains to reach that higher goal. This high level common goal makes it actually easy for us to do this thing [to collaborate, to work together]”*.

There is a great belief in the ability of particle physicist to overcome technical obstacles in order to get their experiments to work. People almost always state with certainty that the LCG will work because they are extremely clever and will make it work. As one interviewee argued: *“Within high energy physics, I think it’s true to say that all people you deal with are highly intelligent and you don’t get people working at this level who are not above average intelligence, given the general population”*. A more significant source of confidence (one might argue arrogance) resides in the belief in the individual skills, competence, creativity and in the context of collaboration. This high degree of competence, the shared goals and internal motivation as well as the collaborative working create a trustworthy environment which drives the community. Trust is considered to be a central element of the collaboration binding them together and, as argued by the physicists, it is crucial in making their collaboration successful as it removes a lot of arguments and discussions in decision-making.

Porra (1999) argues that the culture of collaboration or individualism within a colony is passed on from generation to generation as customs, norms, values, stories and behaviour - its rules of conduct. For particle physicists the collaborative way of working (**rules**) is inherent within individual physicists and this is the only, if we could say, “real rule” that guides them. Being so distributed it is crucial to build a strong sense of community and construct an identity for those involved in Grid’s development in order for the project to function collectively. Going to the pub or going together for lunch when co-located, for example, are one important aspect of this.

There are no rules as one might find in many industries. Formal use of Gantt charts and fixed schedules do not accord with particle physicists’ work. Although, Gantt charts have to be produced in preparation for applying for funding, these serve as a minimal organizing structure for the project. This however, does not mean that they do not believe in planning. To cope with difficulties in this virtual environment, the project has to be flexible and quickly adapt to changes. Even though there is not a clear fixed detailed plan, there is however the plan to carry the project forward by improvising highly pragmatic and practical solutions. This way of working has caused tensions with the computer scientists working in the project who aspire a more plan-based, methodological approach to development: “*[Software engineers] want to design things, they want the project to be very well defined, but (...) by definition physicists normally don’t know what they want. There’s a slight difference in attitude*”.

Another important characteristic of this community is the freedom they have in their work. However, these seemingly spontaneous practices at the individual level are balanced and directed towards the shared object by a level of reflexivity; maintained by continuous and extensive communication between the project’s members. Communication and socialization are seen as crucial for the project to be successful and can take various forms such as meetings of different boards, committees and working groups, virtual meetings, mailing lists, informal face to face meetings on the corridor, semi-structured face-to-face meetings, wikis, blogs. Physicists value face to face meetings and that is why they try to meet with every opportunity: “*Being so dispersed as a development group, we try to have frequent face to face meetings(...)Having the possibility to have technical discussions all together is also good. But having the possibility to have unstructured free time where people can talk to each other is also very, very important. Helps in building the group*”. Such communication is also required for skills development since physicists’ argue that they mostly acquire their skills through word of mouth. Another important form of communication which helps for standardization of the working practices is rotation of expertise.

There is no clear **division of labour** within the collaboration and individuals shift between jobs and have more than one job at the same time. People volunteer to do things and shift between jobs and not because they are forced by someone, but because they want to (though political forces can obviously play a part). Therefore, there is flexibility in roles since people tend to do what they feel they are best at: “*We don’t regard our roles as having fixed boundaries...We tend to do things which we are better at regardless of whose role it might actually be.*”. Furthermore, they would argue there is no strict hierarchy within the collaboration since (as argued by the interviewees), what they argue they have is a collaboration with a spokesperson and volunteers rather than a company with managing director and board. Decision-making is based on discussions were everyone can share their views leading towards common consensus with fights and conflict very rare. Decision-making does not appear to stem from social arbitrariness or political power. This is not to say that politics does not exist, but that they are dispersed, sidelined and the influence of powerful actors is dissipated. There is not a leader that directs what people are doing; rather people have freedom to improvise, to use different techniques and have space for creativity and innovation: As one interviewee who worked closely with Tim Berners-Lee put it: “*Why was the web-invented here? Because Tim had the freedom from this hierarchy to spend a bit of time investigating something which was of interest to him and nobody else here [thought] –*

oh it's a waste of time, never mind. He was working on remote procedure calls. And out of it popped the Web...One guy, sitting in his office, who had a dream."

7 STRATEGIES FOR SUCCESSFUL DISTRIBUTED DEVELOPMENT

Particle physicists acknowledge that the distributed development of the LCG has been a challenging learning journey. Drawing on our Activity Theory framework we observe a number of inner contradictions emerging and being faced throughout the development process. These include i) tensions between particle physicists and computer scientists within the developers group, ii) tensions between developers and users, iii) contradictions between traditional technical models to development and what was actually going on, iv) problems with the extreme size of experiments and the requirements of data analysis compared to previous experiments infrastructures of communication and data storage. Their means of resolving these contradictions provide evidence of their "expansive learning" (in activity theory terms). Such learning drives the progress of the development, and not only gave rise to technological innovations (such as the Grid middleware) and development tools to support distributed collaboration but also to new work practices which have enabled developers to better face the demands of such large-scale distributed development. It is in these practices that we find lessons for those engaged in the distribution of systems development. For this reason we now consider these emergent practices in detail, focusing on the problems faced and how these were resolved.

7.1 Clusters of competence.

Particle physicists are highly influenced by their traditions and past experiences and their practices are rooted in their history and culture. One of the lessons they learnt which led to changes in the way they work was the realization that fully distributed development is difficult because of problems of integrating work effectively: *"It is very difficult to have different teams that are distributed working on the same component because the elements they develop might not work together in the end"*, *"In the past we had joint development of a group of people who were distributed, but that was not working. It was a painful process"*. The disadvantages of fully distributed development were found to outweigh the benefits. For example, management of the project and coordination of work was difficult as the dependencies of the different technical components were too many, something which also made integration difficult and messy. Full distribution of the work, originally believed to be the right way to approach the situation, presented a contradiction to what was actually going on in practice. Their answer to this contradiction was the idea of (in their own words) "Clusters of Competence" which enabled them to structure the development in different competent clusters: *"We are trying to go to a situation where one component is being developed by the same group of people who are all in one place"*, *"We have come up with the idea of strong collaboration between the developers in order to manage the dependencies. Now all the dependencies are such, that there are no conflicts"*. They have created different globally distributed patches of expertise, where experts are co-located, which are then all aligned into a network that facilitates and coordinates the work and the collaboration. Furthermore, this network facilitates communication and sharing of knowledge and expertise among the different clusters: *"The idea here is to have strong collaboration within the developers group as well as collaboration with other countries in order to share knowledge and expertise"*.

Although the clusters of competence were found to provide discipline in messy situations, for the development of some components of the Grid this was not possible and hence there are still people working in a truly distributed way. While virtual communication is important for knowledge sharing and standardization, particle physicists still encourage temporary co-location of developers and therefore establish frequent face-to-face check points since such co-location can resolve future problems, improve awareness, forge understanding and enable

strategic thinking: *“Developers from other countries come here and work for 3 months and this is much more efficient.. It’s much easier to control the activity rather than when people are away. There is still freedom and a level of autonomy but for large features we discuss and decide how to proceed. So there is freedom in a controlled way.”*

7.2 Balancing experimentation with discipline.

All particle physicists must write computer software in order to undertake their physics analysis since packaged applications for this task do not exist. They also have a tradition of large collaborations for developing detectors, accelerators and experiments; however, they do not have formal training in software engineering or traditional systems development: *“As a physicist you do not get much experience in writing software which stays up and is reliable. We work through trial and error and through this you do not get the experience in writing code for stable services”*. They are pragmatic, dirty programmers who like working solutions. They believe in producing things that work quickly and that is why they do not usually prefer the fancy well-thought and well-developed solutions who take time to be developed. Indeed one of the interviewees, jokingly said: *“We are intelligent people; we don’t make bugs so there is no need for methodologies”*. For them developing software is an experimental activity involving trial-and-error in a way similar to the way physics itself is undertaken. However, when asked about this unstructured “experimental” (and risky) way of working, they have all agreed that in such kinds of distributed development projects someone must combine this agility/flexibility together with limited structure/discipline: *“One thing the project learned is that you need management and clear short-term priorities, or else you drift”*. Because this project consists of both particle physicists and computer scientists, some discipline in development activities is seen as needed in order to balance the developers’ individual goals with the shared objective. As one interviewee stated: *“Computer scientists think about technology per se and don’t care about the physics underlying this technology...Computer scientists are not motivated and driven by the same goals as physicists”*. On the other hand, it is also crucial to maintain this flexible and agile character in the way they work in order to quickly adapt and respond to environmental changes: *“We need to formalize the process a bit, but it should allow modifications in a fast pace. Quick feedback from users as well as quick releases for testing is required”*.

7.3 A sense of belonging

Interestingly, when asked what other communities could learn from the distributed development of their Grid, interviewees suggested that what makes their project progress is a combination of factors. As they have argued: *“We’ve many times seen the development of systems by isolated groups involving formal procedures. But these didn’t have good results”*. Therefore something more than co-location and formal procedures is needed in order for such kind of virtual projects to be successful: *“Social is the key really. It makes such a huge difference when people work together for the right reason. True quality comes from within”*. Indeed creating a strong sense of community with shared goals is crucial for their collaboration: *“So it is not so much a software development, the story we have to tell, it is building this community around the grid computing”*. *“Collaboration and building community is really important for distributed development. We work a lot using mailing lists; you can see the different attitude people have before and after they meet in person in those mailing lists”*. Shared goals provide motivation and an identity is constructed for those involved in the development of the Grid: *“If there is a problem that comes up then people will stay up at night, or weekends, and figure out how to get it to work, and it may be a horrible hack, but it will work. There are people out there with a lot of motivation and a lot of knowledge and a lot of skill who will come up with the solutions”*. This sense of community is highly related to the frequency of the face-to-face interactions, the extensive communication flows, timely feedback, keeping all people involved, creating a trustworthy environment, but also depends on the equally important indicators of shared identity such as logoeed pens,

posters, t-shirts worn in conferences etc., and an intense focus on disseminating the project's successes. The feeling of belonging to a group also balances competitive relationships: As they argue: *"Proper management of competition leads to successful outcomes. Without competition brilliant ideas are killed. That is why we work through competing solutions and the best one wins out"*.

In conclusion, particle physicists appear both highly unusual and somewhat traditional in the way they work. Freedom, trust, consensus, charismatic leadership, shared goals and internal motivation are all distinct characteristics of the community and are seen to be its major driving forces. However, this is not to say that politics do not exist or that competition is minimized. Rather, healthy completion exists which helps bring competence in the community and blends expertise.

8 CONCLUSIONS

This paper examined the distributed development of Grid infrastructure in a particle physics setting. Specifically, the focus of concern was to explore the collaborative practices employed by particle physicists in their attempt to develop a usable Grid for their research, and that of other communities, with the aim of offering answers that may be translated into the wider context of global virtual development.

A number of strategies for distributed development were presented based on lessons particle physicists learnt throughout the Grid's development. Particle physicists' collaboration has been described to be exceptional. Their collaborative way of working are rooted in the community's history and culture something which has highly influenced the practices and strategies employed in Grids development. Yet their means of coping appear both unusual and yet somewhat orthodox. In summary the strategies identified were: 1) Structure the development effort in clusters of competence, 2) Encourage temporary co-location of developers, 3) Combine flexibility/agility with structure/discipline, 4) Create a sense of belonging and therefore construct identity for those involved in the development, 5) Facilitate human communication both through virtual means and face-to-face (at least every couple of months), 6) Create a trustworthy environment, 7) Have clear shared goals and rationale. These lessons resonate with many of the trends in management theory around effective distributed working, yet the fact that they emerge from a unique community founded not in existing bureaucratic commercial organisations but in a communitarian science practices provides important evidence to this ongoing debate. Further the study demonstrates the value of employing Activity Theory in researching global systems development practices to explore their inherent contradictions.

Clearly, the unique and obscure nature of the community under study is also a limitation to the study. The practical recommendations provided here should not be seen or treated as prescriptive canonical rules but hopefully they will allow others involved in such distributed virtual development projects to reflect on their practice and their context in light of them.

9 REFERENCES

- Agerfalk, J. P. and B. Fitzgerald (2006). "Flexible and distributed software processes: Old returns in new bowls?" Communications of the ACM 49(10).
- Bajec, M., M. Krisper, et al. (2004). The scenario for constructing flexible, people-focused systems development methodologies. ECIS conference.
- Berman, F., F. Geoffrey, et al. (2003). The Grid: past, present, future. In: Grid Computing-Making the Global Infrastructure a Reality. F. Berman, F. Geoffrey and T. Hey, John Wiley & Sons.

- Bertelsen, O. W. (2000). "Design artifacts: Towards a design-oriented epistemology." Scandinavian journal of information systems **12**: 15-28.
- Bertelsen, O. W. (2003). Contradictions as a tool in IT-design: Some notes, 8th European conference of computer supported cooperative work (ECSCW), Helsinki, Finland.
- Carr, N. (2005). "The End of Corporate Computing." MIT Sloan Management Review **46**(3): 67-73.
- Chetty, M. and R. Buyya (2002). "Weaving computational grids: how analogous are they with electrical grids?" Computing in Science & Engineering **4**(4): 61-71.
- Chompalov, I., J. Genuth, et al. (2002). "The organization of scientific collaborations." Research Policy **31**: 749-767.
- Ciborra, C. (2002). The Labyrinths of Information: Challenging the Wisdom of Systems. Oxford, UK, Oxford University Press.
- Cockburn, A. and J. Highsmith (2001). "Agile software development: The People factor." IEEE Computer
- Colling, D. J. (2002). GridPP - Developing an Operational Grid. UK e-Science All Hands Conference, Sheffield.
- Crawford, K. and E. Hasan (2006). "Demonstrations of the activity theory framework for research in information systems." Australian journal of information systems **13**(2): 49-68.
- Damian, D. and D. Moitra (2006). "Global software development: How far have we come?" IEEE Software.
- De Souza, B. C. and F. D. Redmiles (2003). Opportunities for extending activity theory for studying collaborative software development. 8th European conference of computer-supported cooperative work, Helsinki, Finland.
- Doyle, T. (2005). Meeting the particle physics computing challenge, Public Service Review: Department of Trade and Industry.
- Engestrom, Y. (1987). Learning by expanding: An activity-theoretical approach to developmental research. Helsinki, Orienta-Konsultit.
- EuropeanCommission. (2006). "Future for European Grids: Grids and service-oriented knowledge utilities." ftp://ftp.cordis.europa.eu/pub/ist/docs/grids/ngg3_eg_final.pdf Retrieved 22/02/2007.
- Fitzgerald, B. (2000). "Systems Development Methodologies: The Problem of Tenses." Information Technology & People **13**(3): 13-22.
- Foster, I. and C. Kesselman (2003). The Grid in a Nutshell. In: Grid Resource Management – State of the Art and Future Trends. J. Nabrzyski, J. Schopf and J. Weglarz, Kluwer Academic Publishers.
- Foster, I., C. Kesselman, et al. (2003). The Anatomy of the Grid. In: Grid Computing – Making the Global Infrastructure a Reality. F. Berman, F. Geoffrey and T. Hey, John Wiley & Sons.
- Guy, S. E. (2003). Patterns as artifacts in user-developer collaborative design. 8th European conference of computer-supported cooperative work, Helsinki, Finland.
- Hanseth, O. and E. Monteiro. (1998). "Understanding Information Infrastructure." <http://heim.ifi.uio.no/~oleha/Publications/bok.html> Retrieved 15th of June, 2006.
- Herbsleb, D. J. and A. Mockus (2003). "An empirical study of speed and communication in globally distributed software development." IEEE Transactions on Software Engineering **29**(6): 481-494.
- Herbsleb, D. J. and D. Moitra (2001). "Global software development." IEEE Software.
- Herbsleb, D. J., J. D. Paulish, et al. (2005). Global software development at Siemens: experience from nine projects. International conference on Software engineering (ICSE'05). St. Louis, Missouri, USA, ACM.
- Highsmith, J. (2003). Agile Project Management: Principles and Tools. Cutter Consortium. Arlington, MA.
- Highsmith, J. and A. Cockburn (2001). "Agile software development: The business of innovation." IEEE Computer Society.
- Knorr-Cetina, K. (1999). Epistemic Cultures: How the sciences make knowledge. Cambridge, MA, Harvard University Press.

- Korpela, M., A. Mursu, et al. (2002). "Information systems development as an activity." Computer supported cooperative work **11**(111-128).
- Kuutti, K. (1991). Activity theory and Its Applications to Information Systems Research and Development. Information Systems Research: Contemporary Approaches & Emergent Traditions. H. E. Nissen, H. K. Klein and R. Hirschheim, Amsterdam: North-Holland: 529-550.
- Lacity, M. C. and L. P. Willcocks (2001). Global Information Technology Outsourcing: In Search of Business Advantage, John Wiley & Sons Ltd.
- Lee, G., W. Delone, et al. (2006). "Ambidextrous coping strategies in globally distributed software development projects." Communications of the ACM **49**(10).
- Leontiev, A. N. (1978). Activity, consciousness and personality, Prentice Hall, Englewood Cliffs, NJ.
- Lloyd, S. (2006). From Web to the Grid. PPARC and Parliament and "House" magazine.
- Mwanza, D. (2001). Where theory meets practice: A case for an activity theory based methodology to guide computer system design. Proceedings of INTERACT' 2001: 8th IFIP TC 13 conference on Human-computer Interaction, Tokyo, Japan.
- Nandhakumar, J. and D. Avison (1999). "The fiction of methodological development: a field study of information systems development." Information Technology & People **12**(2): 176-191.
- Nardi, B. (1996). Context and Consciousness: Activity theory and human computer interaction. Cambridge, MIT Publisher.
- Nerur, S., R. Mahapatra, et al. (2005). "Challenges of Migrating to Agile Methodologies." Communications of the ACM **48**(5).
- Orlikowski, W. J. (2002). "Knowing in practice: Enacting collective capability in distributed organizing." Organization Science **13**(3): 249-273.
- Parnas, D. (2006). "Agile methods and global software development (GSD): The wrong solution to an old but real problem." Communications of the ACM **49**(10).
- Pickering, A. (1995). The Mangle of Practice. Chicago, Chicago University Press.
- Russo, N. and E. Stolterman (2000). "Exploring the assumptions underlying information systems methodologies: Their impact on past, present and future ISM research." Information technology and people **13**(4): 313-327.
- Shrum, W., J. Genuth, et al. (2007). Structures of Scientific Collaboration. Cambridge, MA, MIT Press.
- Smarr, L. (2004). Grids in Context. In: The Grid2: blueprint for a new computer Infrastructure. I. Foster and C. Kesselman, Elsevier Inc.
- Traweek, S. (1988). Beamtimes and lifetimes: The world of high energy physics. Cambridge MA, Harvard University Press.
- Vygotsky, L. S. (1978). Mind in society: The development of higher psychological processes. Harvard, Harvard university press.
- Yalaho, A. (2006). A conceptual model of ICT-supported unified process of international outsourcing of software production. 10th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW'06), IEEE.
- Ye, Y. (2005). Dimensions and forms of knowledge collaboration in software development. Proceedings of the 12th Asia-Pacific software engineering conference (APSEC'05).
- Zheng, Y., W. Venters, et al. (2007). "Distributed Development and Scaled Agility: Improvising a Grid for Particle Physics." Under review for International Journal.