doi:10.1111/j.1365-2575.2010.00360.x

Collective agility, paradox and organizational improvisation: the development of a particle physics grid

Yingqin Zheng,* Will Venters† & Tony Cornford†

*De Montfort University, Leicester, UK, email: yzheng@dmu.ac.uk, and †London School of Economics, London, UK, email: w.venters@lse.ac.uk; email: t.cornford@lse.ac.uk

Abstract. *This paper examines systems development in a global collaborative community of high-energy physics and offers insights and implications for agile systems development in other large scale and distributed settings. The paper studies the ongoing construction of the UK's computing grid for particle physics (GridPP), a grid that is itself part of the world's largest grid, the Large Hadron Collider Computing Grid. We observe in this project a collective, agile and distributed performance through which the Grid is constructed. We express this through the concept of 'collective agility' which captures a large distributed performance rather than the more conventional sense of agility as small-group and deliberate systems development practices. The collective agility of GridPP is analysed as a process of 'enacted emergence' expressed through the dynamics of six improvisation paradoxes.*

Keywords: collective agility, improvisation, paradox, grid computing, particle physics

INTRODUCTION

Much recent innovation in systems development has sought to legitimize a more fluid, exploratory and responsive style (Baskerville *et al*., 1992, 2006; Truex *et al*., 2000; Fowler & Highsmith, 2001; Baskerville and Pries-Heje, 2004). These moves away from traditional formalism in systems development methodologies (e.g. DeMarco, 1978; Boehm, 1988) echo the long-standing observation from the field that traditional methodologies are neither effectively nor extensively used (Avgerou & Cornford, 1993; Bansler & Bodker, 1993), but often 'faked' (Parnas & Clements, 1986) and used as a 'fiction' to help create a sense of coherence in day-to-day activities (Nandhakumar & Avison, 1999). Such observations have caused some to rethink the status of method and methodology in systems development. Ciborra (2002), for example, asks us to 'suspend the belief that behind the messy everyday reality there is a geometric universe'. Similarly, Truex and associates argue that 'amethodical' development

(Truex *et al*., 2000) can better appreciate and support innovation and organizational change, adaptation and experimentation, as well as exploiting new opportunities and accidents. If we understand organizational landscapes to be emergent or enacted (Weick, 1993b, 2001), and that technology is created 'in-practice' (Orlikowski, 2000), it makes sense to argue that the way we develop information systems should support a strong contextual contingency and allow for improvisational action and bricolage (Bansler & Havn, 2004).

Agile systems development

Many contemporary systems development practices are oriented towards speed, responsiveness and flexibility. These practices have been given names such as 'high speed software development' (Baskerville *et al*., 2006), 'short-cycle time systems development' (Baskerville and Pries-Heje, 2004), 'web-based system development' (Kautz *et al*., 2007) and most influentially, agile systems development (Fowler & Highsmith, 2001; Highsmith, 2002; Conboy & Fitzgerald, 2004). Agile development has established a large research literature in the past few years, and an extensive following within the practitioner community, based on principles characterized by quickness, lightness and nimbleness (Highsmith, 2002) and on values such as collaboration, communication, simplicity and courage (Beck & Andres, 2005). The practitioner literature includes many versions such as rapid prototyping or quick releases, placing emphasis from predefined procedures, specification and systematic methods (Highsmith, 2002; Williams & Cockburn, 2003; Beck & Andres, 2005). But most studies of agile development still focus on micro-behaviour and related processes in designing and delivering software, and less attention is paid to organizational cultures, institutional conditions and environmental constraints (Abrahamsson *et al*., 2009). This is surprising since, from the days of Brooks' (1979) classic 'The Mythical Man Month' and DeRemer and Kron's (1975) concept of 'programming in the large', it has been acknowledged that systems are developed within an organizational environment, which is as significant in shaping the character of the project and its outcomes as any particular practices. There is thus a need to develop a better understanding of the implementation of agility at the organizational level (Abrahamsson *et al*., 2009) acknowledging explicit linkages to institutional and cultural settings.

Large-scale systems development projects in particular are known to face many challenges as they span institutional and cultural settings. Curtis *et al*. (1988) suggest that the three most salient and interrelated problems they face are thin spread of application domain knowledge; fluctuating and conflicting requirements; and communication and coordination breakdowns. To adopt a style of agile systems development in such settings entails particular difficulties (Reifer *et al*., 2003), including communication, lack of control and lack of trust (Ramesh *et al*., 2006). The strategies proposed to 'scale-up' agile development include developing collaboration tools (Flor, 2006), aligning information technology (IT) components (Lee, Banerjee *et al*., 2006), and managing carefully the balance between flexibility and rigour (Lee, DeLone *et al*., 2006). Yet, few attempts have been made to directly theorize agility and distributed organizational dynamics.

The LCG project

In this paper, we study an emergent form of agile practice within a specific and distinctive large scale and distributed organizational context – the Large Hadron Collider Computing Grid (LCG) – one of the world's largest computing grids. This started development in 2001 and was formally put into use in 2010. The organizational context of the LCG, reflecting that of particle physics (Traweek, 1988; Knorr-Cetina, 1999), is highly distributed with 170 computing centres in 34 countries. Mobilizing to build this Grid was a grand systems development challenge in technical, organizational, political and human terms; an example of large-scale system development on a global basis and one which, as we will argue, seemed to exhibit a quality of agility. Thus, we observed fluid practices that serve as a continuous response to external and internal changes, and continual acts of trial-and-error matched with pragmatic problem-solving approaches. Bricolage and *ad hoc* activities dominate the day-to-day and there is minimal, though vital, use of formal methodologies and centralized control. The people involved, mostly physicists or physicist-programmers, take pride not in methodological rigour but in their pragmatic approach to 'make it work'. As a large distributed project, LCG has faced challenges as mentioned above, and yet their response has not been to employ rationalistic approaches or constraining tool sets as most of the authors above propose, but rather to respond by maintaining their commitment to a flexible and fluid approach – to agility.

The concept of agility

The literature on agility reflects two common approaches. The first sees agility as empirically validated small group methods and practices. The second sees agility as an organizational capability (Sambamurthy *et al*., 2003; Lee, Banerjee *et al*., 2006; Mathiassen and Pries-Heje, 2006), for example, a firms' sense-and-respond capabilities, or dynamic capabilities (Williams & Cockburn, 2003), or the organizational capability to learn, to explore and to exploit knowledge (Overby *et al*., 2006; Mathiassen & Vainio, 2007). We develop a third and distinct perspective, what we call *collective agility* seen as a 'structuring property' (Giddens, 1984) of a collective, instantiated in improvisational behaviour of individuals and groups and in their social interactions. In other words, collective agility is an attribute emergent from the day-to-day practices of social actors. We thus explore agility as a *performance* (Ciborra, 1999; Dyba, 2000).

It is important to make clear the ontological distinction implied by a focus on capability or performance: capability refers to the potential for achievement that an organization has as it draws upon its resources, human, institutional and material, a concept linked to the resource-based view of the firm and core competencies (Wade & Hulland, 2004). In contrast, a performance is an enactment within a context that can create, apply and sustain capabilities. Put it in another way, capabilities are not understood here as something held prior to a performance; rather, they are the medium and outcome of it. The performative ontology (Pickering, 1995) adopted here sees agility as what social actors do when engaging with uncertainty and complexity, and as sustained by collective agency over time and space. Our focus is not a description of agile behaviour or its precursors but on the performance of collective agility that embodies the LCG project. The emphasis is on agility's emergence from disparate practices embedded in the organizational and cultural context. In simpler terms, it is not just that agile system development can be sustained in particular supportive organizational or cultural contexts, but that a context or culture may itself demand (at times) and create a certain type of agile performance.

Organizational improvisation and paradoxes

To unpack the complexities of collective agility as an organizational performance, we draw upon the literature of organizational improvisation (Weick, 1998; Cunha *et al*., 1999) with a focus on collective, collaborative and coordinated improvisational activity. Improvisation is essentially an individual or small group practice, immediate and situated, whereas the agility we study is that which emerges from a collective performance – an organizational improvisation. This literature is reviewed and organized from the perspective of paradox (Mirvis, 1998). It has been frequently noted that innovations such as short cycle time development or agile methods involve tensions and paradoxical elements, for example, learning to 'plan not to plan' (Baskerville, 2006) or to achieve a 'disciplined messiness' (Highsmith, 2002) and Baskerville's (2006) calls for a rejection of polar distinctions between concepts like planning and serendipity, or discipline and creativity. The concept of paradox is not intended to suggest logical impossibility or irresolvable conflict; rather, paradox provides a means of presenting and analysing productive tensions, dynamics and motivating challenges of systems development. In constructing a set of improvisation paradoxes and applying them in the analysis, we reveal the embracing and balancing of such paradoxical elements as a key to understanding agility within distributed collaborative system development.

In summary, this paper introduces collective agility, a concept developed from a paradoxical perspective, to describe a particular genre of organizational performance. The rest of the paper is organized as follows. The next section further develops the conceptual constructs, i.e. the improvisation paradoxes, from the literature of organizational improvisation. Research methodology and case description can be found in the succeeding sections. The next section, Enacting Paradoxes, presents an in-depth analysis of the case using the improvisation paradoxes, extended to discussions and implications in the following section. The last section concludes the article.

IMPROVISATION PARADOXES AND ENACTED EMERGENCE

Existing research on organizational improvisation mostly considers it as a creative group performance with little formal planning and minimal central control, like a jazz performance (Barrett, 1998; Hatch, 1999) or improvisational theatre (Crossan, 1998). Cunha *et al*. (1999) define improvisation as 'the conception of action as it unfolds, by an organization and/or its members, drawing on available material, cognitive, affective and social resources'. This definition emphasizes two aspects. First, the convergence in time of conception and execution (Moorman & Miner, 1998), or 'real-time planning' (Miner *et al*., 2001). This resonates strongly with the basic notion of agility as quickness, lightness and nimbleness (Highsmith, 2002). Second, bricolage – the aspect of finding solutions from available rather than optimal resources – which is often implied or used interchangeably with improvisation (Weick, 1993a,b; Ciborra, 2002).

Within the field of information systems, ideas of improvisation and bricolage have often been used to critique the dominant ontology of planning and control and the pervasive normative tendencies that follow (Orlikowski, 1996; Ciborra, 1999; 2002; Lanzara, 1999). Organizational improvisation literature does not deny or negate the value of such concepts, but suggests that it is in the tension and interaction between these and their opposites: structure and change, order and chaos, control and freedom, that creative attitudes, innovative outcomes and productive practices may be found. The performative view of agility adopted here reveals the 'tensions and oppositions between well-founded, well-reasoned and well-supported alternative explanations of the same phenomenon' (Poole & Van de Ven, 1989). Thus, we draw on the established tradition of paradox as a dialectical device to examine complex situations and to build theory (Poole & Van de Ven, 1989; Lewis, 2000; Smith & Tushman, 2005). Lewis (2000) describes three categories of paradoxes prevalent in organizational studies – learning (old/new), organizing (control/flexibility) and belonging (self/other). We use these three categories to synthesize the literature on organizational improvisation and propose a set of improvisation-paradoxes. Table 1 presents the six constructed improvisation paradoxes with examples of the concepts they are based on. These concepts are highlighted in italic in the following elaboration of the improvisation paradoxes.

Paradoxes of learning

Paradoxes of learning arise from the tension between *old* and *new*, 'the struggle between the comfort of the past and the uncertainty of the future', which are fundamental to processes of innovation, transformation and sensemaking (Lewis, 2000). On this basis, we identify two pairs of paradoxes of learning: *Learned Improvisation*, i.e. improvisation drawing on past experience and situated within environmental constraints; and *Reflective Spontaneity*, making sense by ex-post interpretation and rationalization.

Learned improvisation

This paradox is related to the tension between the immediate (the here-and-now environment and context) and the historic (the understood, interpreted, documented and remembered past). 'Learning requires using, critiquing and often destroying past understandings and practices to construct new and more complicated frames of references' (Lewis, 2000). For example, jazz musicians recall music that has been performed and learn from it (Berliner, 1994). Improvisation is often a response to *task uncertainty*, and *environmental turbulence* (Moorman & Miner, 1998). Unexpected and 'unplanned-for' (Miner *et al*., 2001) occurrences or tasks can arise

Table 1. Tensions and paradoxes in organizational improvisation

	Paradoxes of learning	
	Immediate	Historic
Learned improvisation	Environmental turbulence (Moorman & Miner, 1998, Ciborra, 1996) Task uncertainty (Miner *et al*., 2001) Task complexity (Weick & Roberts, 1993; Hutchins, 1995)	Organizational memory (Ackerman and Halverson, 1998; Moorman & Miner, 1998; Weick, 1998) Routines (Weick & Roberts, 1993; Hutchins, 1995) Practicing (Moorman & Miner, 1998; Weick, 1998)
	Spontaneity	Reflexivity
Reflective spontaneity	Convergence of planning and execution (Moorman & Miner, 1998) Drop your tools (Weick, 1993a) Trial-and-error, *bricolage* (Lanzara, 1999)	Retrospective sense-making (Weick, 1993b) *Ex-post* interpretation (Lanzara, 1999) Transient constructs (Lanzara, 1999)
	Paradoxes of organizing	
	Unfolding	Planning
Planned agility	'Unfolding ontology' (Knorr-Cetina, 1999) 'Unfolding circumstances' (Ciborra, 1999) 'The spontaneous' (Weick, 1998) Drifting (Ciborra, *et al*., 2000), Flow (Hatch, 1999)	Visions (Mintzberg & McHugh, 1985, Hutchins, 1991, Weick, 1993b, Hatch, 1999) Plan to improvise (Miner *et al*., 2001) Artful planning (Baskerville, 2006) A sense of urgency (Hutchins, 1991, Crossan, 1998, Mirvis, 1998)
	Practices	Structure
Structured chaos	Organized anarchy (Cohen *et al*., 1972) Knowing in practice (Orlikowski, 2000) Fractures, discontinuities, inconsistencies (Lanzara, 1999) Ambiguity (Hatch, 1999)	Minimal structure (Cunha *et al*., 1999) Collateral structure (Cunha *et al*., 1999) Aesthetic of imperfection (Weick, 1999) 'Experimental culture' (Cunha *et al*., 1999) Pro-innovation culture (Mirvis, 1998; Weick, 1998; Miner, *et al*., 2001)
	Paradoxes of Belonging	
	Individuals	Collectivity
Collective individuality	Individual skills (Brown & Duguid, 1991; Mirvis, 1998) Creativity (Barrett, 1998; Hatch, 1999; Kamoche, *et al*., 2003)	*Group cohesion* (Weick & Roberts, 1993; Hutchins, 1995) Facilitative leadership (Crossan, 1998) Trust and kinship (Weick, 1993a, Crossan, 1998) Fluid communication (Orlikowski, 1996, Miner *et al*., 2001)
	Anxiety	Confidence
Anxious confidence	Anxiety (Mirvis, 1998; Cunha, *et al*., 1999) Moods (Ciborra, 2001) Emotionality (Hatch, 1999) Sense of urgency (Hutchins, 1991, Crossan, 1998, Mirvis, 1998)	Individual skills (Brown & Duguid, 1991; Mirvis, 1998) Aesthetic of imperfection (Weick, 1999) Prideful wariness (LaPorte, 1996)

inside the collective too (Cunha *et al*., 1999), for example, when *task complexity* seems to be beyond the scope of rational planning, accumulated knowledge or predetermined method (Hutchins, 1995). Organizational improvisation can also be linked to *deliberate innovation* – for example, visions which articulate a gap between reality and possibility can induce actions which are partly planned yet significantly emergent (Mintzberg & McHugh, 1985) and improvised (Crossan *et al*., 1996).

To cope with uncertainties and complexities of the environment, context or task, people need to draw upon a repertoire of *organizational memories*, 'learned ways of thinking and behaving' (Moorman & Miner, 1998). The paradox of learned improvisation thus also reflects the tension between the reliance on 'habits of thought' and routines (Weick & Roberts, 1993; Hutchins, 1995) and a will to depart from organizational traditions and norms (Cunha *et al*., 1999) – to *drop your tools* (Weick, 1993a). This balance is intricate, as successful improvisations are often based on accumulated knowledge and experience from extensive *practicing* in the past (Moorman & Miner, 1998). The construct of 'history' here is also related to collective understanding and organizational culture, e.g. in our case, the experimental culture of particle physics.

Reflective spontaneity

'Reflective spontaneity' is a paradox expressing *ad hoc* experimentation (spontaneity) and *post hoc* recovering of rationalization by the collective, Weick's *retrospective sensemaking* (Weick, 1993a). Improvisers often have no choice but to engage with the situation with no time for thorough reflection, and the significance of the action is often only (re)discovered after the event. As Weick (1998) suggests, 'to improve memory is to gain retrospective access to a greater range of resources'. Retrospective sensemaking (Weick, 1993a) can thus provide order, purpose and coherence (Barrett, 1998) to practices of *trial-and-error* or *bricolage* (Lanzara, 1999). Meaning can arise from *ex-post interpretation* and sensemaking by a number of dispersed agents, rather than from *ex ante* planning and implementation by a central designer (ibid.). Such sensemaking can be facilitated by *transient constructs* (ibid.), such as 'makeshift artefacts, recombinant routines . . . ephemeral organizations, disposable symbols, fugitive meanings' than can sustain some continuity and stability. Milestones and deadlines, for example, may serve these purposes. Thus, at the macro-level, an unfolding improvisational performance and the reflections on it, give rise to an *emergent order* (Miner *et al*., 2001) which in turn can be drawn upon by others (Orlikowski, 2000).

Paradoxes of organizing

Paradoxes of organizing reflect tensions between control and flexibility (Lewis, 2000), formal and informal, integration and differentiation (Chae & Bloodgood, 2006), denoting 'an ongoing process of equilibrating opposing forces that encourage commitment, trust and creativity while maintaining efficiency, discipline and order' (Lewis, 2000). Two paradoxes from the literature are summarized here as *Planned Agility* and *Structured Chaos*. The former underlines the tension between the deliberate action of planning and the uncontrolled processes of drifting and unfolding; the latter refers to the tension between chaotic day-to-day practices and minimal structures.

Planned agility

As Weick (1998) puts it, 'improvisation is a mixture of the pre-composed and *the spontaneous*'. Miner *et al*. (2001) suggests that organizations can plan to improvise and routinize processes to stimulate improvisation, without the actual content of the improvisation being planned in advance. This is related to what Baskerville (2006) refers to as *artful planning*. Degrees of planning for improvisation encompasses two aspects: clearly articulated goals (Crossan *et al*., 1996; Orlikowski, 1996; Barrett, 1998), and milestones and action deadlines (Cunha *et al*., 1999). Clearly articulated goals can provide a sense of direction and shared *vision*, often operating via culture or ideology (Weick, 1993b; Mintzberg, 1995), and serve as a 'magnetic field' which, without prescribing individual action, is strongly normative in shaping such action (Cunha *et al*., 1999). Short-term milestones and deadlines build a *sense of momentum and urgency* (e.g. Hutchins, 1991; Crossan, 1998; Mirvis, 1998) and sustain a 'state of *flow*' (Hatch, 1999). They provide opportunities to keep track of the variations between dispersed innovative actions and priorities within the collective goal. In other words, even though day-to-day practices may be unplanned, *ad hoc* and *drifting* (Ciborra *et al*., 2000), minimal strategic planning and management can ensure that this is oriented towards the goal.

Structured chaos

Organizational improvisation might be seen as a form of *'organized anarchy'* characterized by problematic preferences, unclear technology and fractured participation (Cohen *et al*., 1972; Hutchins, 1991). Cunha *et al*. (1999) suggest '*minimal structure*' to express the controls desired to achieve improvisations that progress (Orlikowski, 1996; Crossan, 1998; Weick, 1998). Minimal structure refers to a shared sense of rules, norms and identity among members of a community of practice (Brown & Duguid, 1991), and which can be drawn upon by members to mediate their *knowing-in-practice* (Orlikowski, 2000), yet allow them to depart from canonical practices and initiate changes. A *collateral structure* provides non-intrusive support to learning communities allowing space for fluid and interpretative practices to take place across boundaries of groups (Cunha *et al*., 1999).

Minimal and collateral structures allow the cultivation of an *experimental culture* (Cunha *et al*., 1999) or *pro-innovation culture* (Mirvis, 1998; Weick, 1998; Miner *et al*., 2001), which nurture individuality through features such as tolerance to error (Barrett, 1998; Crossan, 1998; Hatch, 1999). Weick proposes an '*aesthetic of imperfection*' as an important condition for improvisation, based on an 'estimate of the degree of organization and form that could have been extracted retrospectively from the materials at hand, given that they were generated by a fallible human being acting publicly under time pressure, with fallible tools' (1999). Lanzara (1999) similarly talks of *fractures, discontinuities, inconsistencies*, deviations from current routines and puzzling or random behaviours in innovative processes. Yet, imperfection and murkiness can embody evolutionary opportunities for novel practices and forms, and lead to further productive combinations and transformations.

Paradoxes of belonging

Paradoxes of belonging emerge ‘because actors strive for both self-expression and collective affiliation’ (Lewis, 2000). This tension is particularly distinctive in improvisational activity, because by nature, members of an improvisational collective tend to be self-driven, intelligent and creative people, yet they also have an acute appreciation that success relies on collaborative effort. It is through trust and mutual support that they acquire confidence and strength in face of pressure and challenges. We adopt under this category Mirvis’ (1998) *Collective Individuality* and *Anxious Confidence*, enriching them by linking them to organizational improvisation literature and theory.

Collective individuality

Creativity and *individual skills* (Barrett, 1998; Hatch, 1999; Kamoche *et al*., 2003) may be encouraged and supported, but individual freedom is inevitably bound by a level of *group cohesion* in order to achieve a collective goal, especially when task complexity is beyond the cognitive capacity of any individual (Weick & Roberts, 1993; Hutchins, 1995). As Weick (1998) puts it, ‘discussions of improvisation in groups are built on images of call and response, give and take, transitions, exchange, complementing, negotiating a shared sense of the beat, offering harmonic possibilities to someone else, preserving continuity of mood, and cross-fertilization’. *Facilitative leadership* (Barrett, 1998; Crossan, 1998), *trust* (Weick, 1993a; Crossan, 1998) and *fluid communication* (Orlikowski, 1996; Miner *et al*., 2001) nurture group performance. Such emotional ties do not have to stem from self-disclosed intimacy but from shared actions, ‘hanging out’ and a sense of membership in the collective (Barrett, 1998).

Anxious confidence

Emotional ties also serve to provide a ‘safety net’ for members of a collective to cope with anxiety, or to deal with the affective element in their performance (Cunha *et al*., 1999). Ciborra (2001) considers improvisation itself as a mood and contrasts it with conventional moods of the systems development context such as panic or boredom, both of which fog vision and conceal possibilities for action. Mirvis (1998) suggests ‘anxious confidence’ as the means to live with the ambiguity, complexity, and challenges of working in an improvisational collective. Similarly, LaPorte (1996) (cited by Weick *et al*., 1999) speaks of ‘prideful wariness’ when discussing air traffic controllers. While Mirvis focuses mostly on individual capability and confidence, confidence is not only experienced through individual knowledge and skills (Hutchins, 1991; Orlikowski, 1996; Moorman & Miner, 1998) but also in aspects of organizational cultures, such as a history of innovation and 'aesthetics of imperfection' (Weick, 1999), which can be drawn upon as 'learned ways of thinking and behaving' (Moorman & Miner, 1998).

Paradoxes and enacted emergence

The improvisation paradoxes explored above embody a sense of tensions found in an agile performance, particularly in a distributed context. To capture the dynamic duality, we adopt two of Poole and Van de Ven's (1989) four modes of working with paradoxes – to first accept the paradox and use them constructively, and then introduce a new term or concept to resolve the paradox. In this spirit, we propose the term 'enacted emergence' to portray a paradoxical and agile performance that is both constructive and emergent.

Indeed, it is often pointed out that information systems development is an emergent socio-technical activity (Baskerville and Pries-Heje, 2004; Orlikowski, 1996; Truex & Baskerville, 1998; Chae & Poole, 2005). The improvisation paradoxes developed here juxtapose and reveal the seemingly opposite elements of such improvisation, and can reveal the tension between environment and history, spontaneity and reflexivity, unfolding and planning, practices and structure, individual and collective, and anxiety and confidence. These elements are bound together in a constant mutual constitution. Enacting elements on one side of Table 1 give rise to elements on the other side, for example, high-level planning, direction and minimal structure provides the support, freedom and safety net for people to explore through trial-and-error, improvise and innovate. Seen the other way, seemingly disorderly and chaotic day-to-day practices can produce order, direction and meaning through retrospective sensemaking. Individuals encouraged to embrace their individuality and thinking free cultivate a culture of democratic meritocracy, while a high level of creativity and competence, as well as common goals can inspire trust, commitment and voluntarism.

Collective agility is then a phenomenon of enacted emergence in the sense that, while rooted in creative human agency, i.e. the improvisational practices (including the dimensions of planning, organizing and structuring) of knowledgeable and reflective social actors (individuals and teams), agility is an attribute of the distributed collective that emerges from the paradoxes and exists as a combination of intended and unintended consequences of these activity. In the following sections, we examine in detail how the particle physics community enact these improvisation paradoxes and sustain a level of collective agility.

RESEARCH METHODOLOGY

The Large Hadron Collider Computing Grid (LCG) provides a distinctive case of distributed systems development (Venters & Cornford, 2006). This work focuses on the UK's component of the project – the UK Participle Physics Grid (GridPP). Core data collection took place from 2006 to 2008 and included participant observations of weekly meetings, various UK and international GridPP workshops and meetings and relevant conferences. We had full access to the GridPP main documentation, and subscribed to its main mailing lists.

Table 2. Details of research activities

Research methods		Examples	Data collection
Semi-structured interviews		Members of GridPP, middleware developers, members of LCG at CERN, physicist users . . .	Audio-recorded, transcribed, coded
Participant observations	Virtual meetings	weekly GridPP PMB meetings weekly deployment team meetings	Audio-recorded, notes taken, not transcribed
	Face-to-face meetings	GridPP collaboration meetings, PMB face-to-face meetings, deployment team face-to-face meetings,	Many audio-recorded, notes taken, not transcribed
	Site visits	GridPP site readiness review	Notes taken
Secondary data		GridPP publications, GridPP documents, GridPP website, wiki, blogs, mailing lists	Frequent consultation

Table 3. Details of interviews

Roles of interviewees	Number	Notes
GridPP PMB members	12	Including project leaders, representatives of all other major boards, and liaisons with other partners.
GridPP technical experts	15	e.g. Tier 1 manager, Tier 2 manager, technical coordinator, deployment, sys-admins, other software developers
Active physicists	9	Often overlapping with other roles
Middleware developers	5	Based in the UK and CERN
LCG technical experts	11	e.g. LCG Grid deployment, experiment integrator, other software developers

Forty eight semi-structured qualitative interviews of between one and one and a half hour were undertaken at universities across the UK and at CERN (European Organisation for Nuclear Research) in Geneva. Table 2 provides details of the research activities undertaken while Table 3 shows a summary of the principal interviewees. Sampling was based on functional groups of GridPP, starting from the Project Management Board (PMB) for an overview, then proceeding to representatives from the three main activity areas: applications, middleware and infrastructure. Another group of interviewees were users from the LHC experiments, many of whom were involved in some development activities. We interviewed managers and technical experts of the LCG at CERN to contextualize our observations. This showed that the practices of GridPP were not unique to the UK but have roots in an international particle physicist culture. When necessary, interviewees were revisited. Interviews were audio-recorded, transcribed and coded for analysis using the Atlas.Ti software (Scientific Software Development GmbH, Berlin, Germany), though not rigidly so as to avoid being restricted by the software. Data analysis was closely integrated with theoretical development in an iterative process, one feeding into the other.

We identify three stages of data analysis. The first was open coding of the data, labelling aspects of the project, practices, and emerging ideas (Table 4 shows an example). This exercise, combined with the embedded understanding acquired by the researchers from secondary material and during participant observation, provided an appreciation of the com-

Table 4. Example of quotations and coding

Quotations	Interviewee	Codes
I'm trying not to use the word senior to imply there's a real hierarchy. I mean people get promoted to be professor or whatever but it really hasn't nothing to do with the way it works, okay? That's internal to the university. So um, those people that you know, formally might seem more senior, this is relevant, their peers with people like [XXXX] and you know, really rely on people like that to make it work technically. So they're fully trusted to just get on with it in the deployment board. Okay? So it's a fairly flat structure really. There's no, there's no company-like structure of management board sets policy and another group sets something else and then you know, down the bottom, people do what they're told. It's nothing like that at all.	Member of Project Management Board	[collaboration] [democratic meritocracy] [flat structure] [mutual respect]
So I was going to come at it from the physicists' point of view to start with because it's very important for the physicists because there's so many things that they have to do in order to be able to interpret something that's been true in the data, that they have to trust what other people have done. And this is even more so when you have such big detectors as the LHC ones.	GridPP technical expert (based in the UK)	[trust] [PP history and culture]
And if you go into these big bang mode where it takes you two years to put this into production, particularly in a community which is as unstable as this is, as uncertain as this is, where changing the beam, which is something totally out of our control, can have implications everywhere, this is not the right policy. And you have to be much more agile in the trends in software engineering and agile in programming, and I am sure you know all about that. And here you do have to do that.	LCG technical expert (based at CERN)	[agility] [pragmatism]

plexity of the project and gave a sense of the tensions inherent in such work. For example, experimental physicists were not always keen to follow procedures. Similarly, while there might seem to be prevalent adhocracy and frequent fire-fighting, the project was unified in their confidence that the system would work. With these broad ideas in mind, our theoretical exploration led us to the literature of organizational improvisation, which has a strong resonance with the data, and already entails a paradoxical dimension (Weick, 1998). This process gives rise to a draft analytical framework of improvisation paradoxes.

In the second round of data analysis, we used the conceptual constructs of the improvisation paradoxes as categories to set up and iteratively refine code families in a way similar to axial coding in grounded theory (Corbin & Strauss, 2008). These codes were presented in a network view, and relationships between the codes were identified. But relationships were not understood as indicating causality. These 'networks' were verified and modified against further observations and interviews. This was an iterative process until the key conceptual constructs were sufficiently refined and saturated. We verified our findings with a survey, not reported here, which largely confirmed the themes.

In summary, the analysis reported here is the result of iterative reflections and ongoing discussions within the research team and with GridPP members, rather than a narrow machine-derived account – our own engagement with organizational improvisation and sensemaking. While all the quotes given here are taken from interview transcripts, the ideas have also been significantly reinforced by informal conversations and participant observations.

This is not to say that the GridPP community is unified in their opinions. Tensions, conflicts and different views are inevitable in any undertaking of this scale. Nevertheless, the research attempts to capture the distinctive features of GridPP, and this account has been broadly supported by three GridPP PMB members who were presented with the key findings of this paper.

THE PARTICLE PHYSICS GRID

In April 2010, the Large Hadron Collider (LHC) particle accelerator at CERN, the European Laboratory for Particle Physics, started again after a problematic public launch in 2009. The LHC collides hadron particles at energies close to those of the Big Bang in a search for the elusive 'Higgs-Boson' particle believed to be responsible for matter having mass. These collisions will produce data for the four LHC experiments. Since the Higgs-Boson is conjectured to be extremely difficult to find, likened to searching for a 'needle in 20 million haystacks', the number of collisions and the subsequent data produced, is vast. The LHC envisages producing 15 million gigabytes of data a year – equivalent to a digital videodisc every 15 seconds or 1% of 2006 global information production (Lee, Banerjee *et al*., 2006). To store and analyse this data, the LHC requires the equivalent of 100 000 personal computers spread across the globe and working as a Grid (Britton *et al*., 2004).

A grid from a technical perspective is a computing platform for coordinated resource sharing and problem solving suitable in data-intensive and computer-intensive applications (Foster *et al*., 2001). A grid connects and coordinates diverse distributed and heterogeneous computing resources, presenting itself to users as though it was a single resource.

The GridPP project started in 2001 and has two main activities: developing software to allow users to submit computing jobs to the LCG, and developing and operating the UK's component of LCG. GridPP is involved in developing applications and middleware (the grid's 'operating system') as well as providing technical infrastructure including storage and processing units. As shown in Figure 1, the LCG has a hierarchically tiered structure, with Tier 0 at CERN; Tier 1s consisting of the national IT centres in each of the major countries involved in the project; and Tier 2s being the regional centres in each country. GridPP consists of the Rutherford Appleton Laboratory as the Tier 1 centre, and four Tier 2 centres, each coordinating a number of institutes in their region.

GridPP is managed, as with the wider LCG, by what one interviewee described as a 'democratic meritocracy'. Figure 2 shows GridPP's management structure which is best described as a network than any sort of hierarchy. The PMB is the heart of the network coordinating the project. It provides quarterly reports to the collaboration board which consists

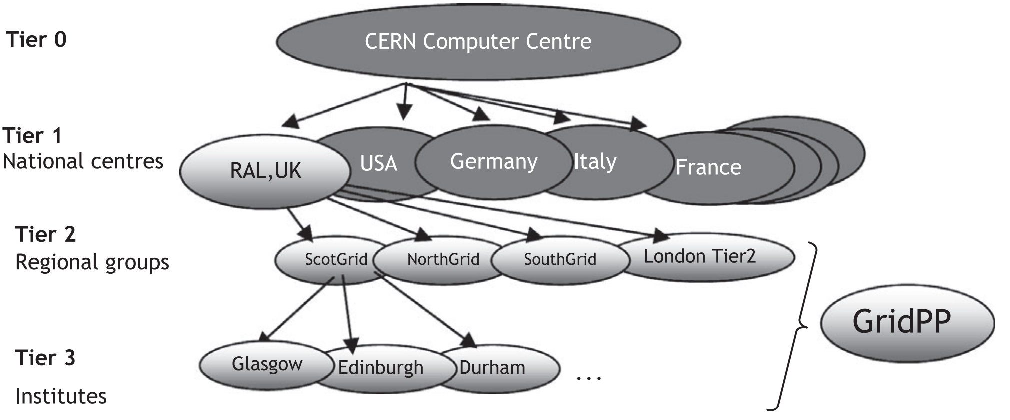

Figure 1. Infrastructure of LCG (Including GridPP).

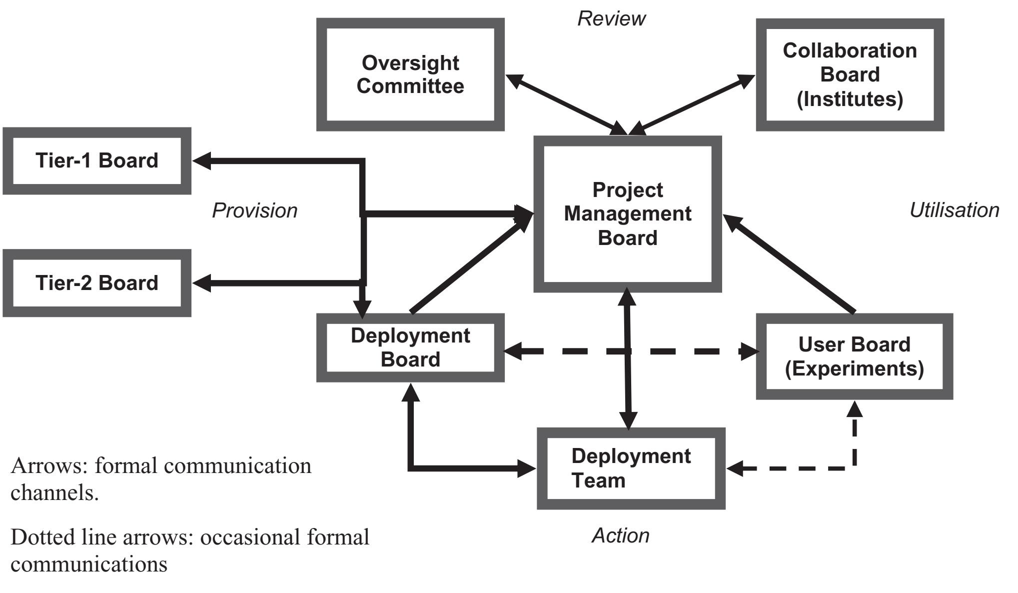

Figure 2. Organizational chart of GridPP (Adapted graph from the GridPP website).

of representatives from the 19 institutes. The participating institutes enter the collaboration not under any legal obligation, but bound by a memorandum of understanding which specifies the amount of resources and the level of service that each site is expected to provide, and the funding and support they will receive from GridPP in return. This document serves as a 'gentlemen's agreement' and there are no formal lines of authority between GridPP and the member institutes other than this collaborative relationship. Decisions are made on a democratic or consensual basis and implemented by influence and persuasion.

Developing LCG has been seen from the start as a highly distributed, complex and poorly defined systems development challenge. Cutting edge hardware and software is used, new software standards have to be negotiated, and middleware along with a wide range of supporting software, developed in a range of countries and programming languages. LCG is developed with close involvement of members of the user community who exerts strong influence and pressure for the completion of a working system, which has to be achieved with limited time and resources.

The system development practices used within GridPP broadly coincide with the general principles of agile methods; ‘individuals and interactions over processes and tools; working software over comprehensive documentation; customer collaboration over contract negotiation; and responding to change over following a plan’ (Fowler & Highsmith, 2001). A technical expert with experience of GridPP described it as a ‘bottom-up approach’.

The systems development practices observed in this case are similar to those described by Baskerville and Pries-Heje (2004) as ‘short cycle time systems development’. Table 5 compares similarities and differences between the practices observed in the two studies. The final column of the table indicates underlying organizational implications related to the identified practices. Beyond these similarities with other agile or short-cycle development projects, some challenges and characteristics of LCG and GridPP are distinctive, in particular, the scale of the system and the distributed nature of its own environment raises demands for scalability and interoperability. For example, LCG draws on several regional grids in Europe, North America and Scandinavia each using different middleware stacks. Within the European project, the middleware is modularized and its components developed in a variety of programming languages. Middleware releases are tested in small-scale pre-production systems but they tend to be problematic when implemented across the whole system. The Grid therefore evolves as advanced users actively engage in using, testing and reporting problems. System development cycles are not only simultaneous or overlapping activities of development, testing and use, but also include complete parallel solutions which compete with each other. Finally, there are tensions around whether the Grid should be generic enough for other communities of users (which it is in part funded to be), or whether it should be tailored to particle physics (the main users and developers); and tensions between the powers of system administrators of local sites, who might wish to prioritize the needs of their local institute and the requirements of the LHC experiments.

ENACTING PARADOXES

The enactment perspective proposed by Weick (2001) suggests that organizations ‘construct’ their environment before they ‘respond’ to it or try to control it, and this can be understood as a process of interacting and sensemaking. Enactment embodies this sense of action and of creation. In this case, the particle physicists ‘reconstruct’ the task of building a new distributed

Table 5. Comparing characteristics of systems development practices with those of 'short cycle time system development' presented by Baskerville and Pries-Heje (2004)

Short cycle time system development	Compared to system development practices in our case	Organizational implications
Causes		
Vague requirement	Yes. Vague requirements because it involves new technology and new experiments.	A collective attitude to deal with uncertainty and ambiguity;
Lack of experience	Yes	
Time pressure	Yes	Capability of organizational learning;
	Other causes:	
	Faced with enormous uncertainties and environmental turbulence.	Capability to work under great pressure;
	Scale	Distributed management
	Existing culture of the particle physics community favours exploration, trial-and-error and bricolage.	Drawing upon organizational memories
System development practices		
Prototyping	Yes. 'rapid prototyping' Result: documentation cannot catch up with the speed of changes.	Exploration, spontaneity
Release orientation	Yes, 'fast development', 'nightly build' and 'monthly release'	Incremental changes
Tailored methods	Yes, or no explicit use of methodology or methods.	Flexibility
Coding your way out	Yes, 'hacking'	Pragmatism
Parallel development	Yes	Coordination, negotiation, persuasion
Fixed architecture	No. Driven by user requirements, which also evolve.	A common goal and shared vision
Components based development and use	Yes. Particularly necessary due to the distributed model.	Coordination
Tool dependence	Yes but mostly self-developed.	
Dependence on good people	Yes, very much so.	Democratic meritocracy, weak authority, high autonomy
Customer involvement	Yes. Power users use and test the system from very early on. The experiments develop applications to run on the Grid, with heavy interactions. Developers select power users as guinea pigs, and cultivate their user communities.	Learning, community building, informal communications
Maintenance ignored	No, but it is problematic.	
Quality is negotiable	Yes	'Aesthetics of imperfection', pragmatism
	Other practices: Parallel solutions competing against each other.	Federated structure

technology as one that they are largely familiar with – a distributed experimental collaboration – and it is seen as just another task that they have to complete in order to achieve the shared goal – doing new physics.

As introduced above, we frame the systems development activity in GridPP as an organizational improvisation that is animated by various tensions. We present our analysis, drawing on improvisation paradoxes but in a slightly different sequence to Lewis. In this case, we see the nature of the grid development as being fundamentally driven by a sense of *belonging* and start with this concept. We then move to paradoxes of *organizing* and conclude with *learning*.

Belonging to GridPP

There is a sense of a strong community bond among GridPP members, which we express in the concept of *collective individuality* (Tables 1 & 6). Most members of GridPP are particle physicists or have a physics background. One consequence is that members are motivated by both a shared history and a shared goal. This goal is not to build a grid, but to discover new physics. They work for the same vision despite strong competition between similar experiments. As commented by one interviewee

> I said I was proud of being a particle physicist, this is because particle physicists always get the job done; by and large because they are driven by one fundamental thing. They want their experiment to work when the beam gets into the accelerator, okay? And that transcends everything else they do.

Coupled with the shared goal is a high level of trust as shown very clearly from the interviews:

> Everyone trusts each other to be doing the best they can . . . That fundamental trust drives our particle physics group.

Table 6. Paradoxes of belonging in GridPP

Individuality	Collective individuality	Collectivity
– Intelligence – Autonomous – Freedom at work – Improvisation	Community bonds among free-thinking individuals	– Shared goal of physics – Emphasis on hanging out – Facilitative leadership – High level of trust – Hanging out
Anxiety	**Anxious confidence**	**Confidence**
– Uncertainties – Unreliable software – Pressure from CERN and from users – Funding shortage	Confidence as a capability to handle anxiety	– Cleverness – 'It will work' – History/organizational memory – Aesthetic of imperfection

> You have to trust that people will step up . . . and do the dirty work as well as doing the glamorous work.

Particle physicists have been encultured above all to respect intellectual capacity (Traweek, 1988). With a high level of trust, people generally enjoy a high level of autonomy at work, usually without clear instructions or strict supervision. Individuals will try to solve a problem, develop software, write a document, not because their line manager told them to, but because they felt that it was something useful to do. Individuals are driven by individual motivations – but they also desire that their contribution be recognized as valuable. Key technologies in particle physics have emerged in this manner – with the main analysis system for storing particle physics events (called 'Root') began with one developer trying to solve a problem in a new way without institutional support, similar to their development of the Web. As one senior CERN employee who shared an office with Tim Berners-Lee recounted:

> 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 said – 'oh it's a waste of time, never mind'. He was working on remote procedure calls, and out of it popped the web.

With members based in disparate institutes, it is important to develop social and emotional bonds among individual members. The deployment team provides a good example.

> We have to work very well together as a team, in order for GridPP to be successful. And . . . it's quite a complicated structure – there are multiple channels of communication, some of which are duplicated some of which are contradictory, and there are all sorts of ways in which information flows. And anything that you can do to oil the cogs of the machine is going to help. . . . And I think for us to socialise together is a very important thing.

'Going to the pub' when and wherever they meet is one aspect of this since it 'fosters a bond' between people and allows them to discuss their frustrations caused by the size and complexity of the project. During such social occasions, work is invariably discussed, people 'let off steam' and negotiations are made. These social meetings are sporadic and between them the constant communication through video conferencing, email, messaging lists, blogs and instant messaging continue. Relationships develop between parties who have never met. Many attend the regular videoconference meetings simply to get a feel for the 'mood' and a sense of connection – often having the meeting running on their computer while undertaking other tasks.

Communication supports a mood of *anxious confidence* (Tables 1 & 6), that mediates the pressure of the LHC switch-on and of showing the UK in a good light among the worldwide particle physics community. Along the way, GridPP has to face many unplanned-for occurrences and environmental turbulence in funding, human resources, external and internal technological changes, hardware and software configurations, technical requirements from the experiments, computer market conditions and other institutional and political factors. Indeed, the project is 'committed to something that it is not quite funded' (PMB member) and in March 2007 were allocated only 70% of the anticipated funding for Phase 3 (2007 to 2011), which resulted in support posts being cut. Nevertheless, the collaboration remains committed, engaged and always 'just about' on top of things. They may appear to be constantly fire fighting, discovering problems, managing crises and negotiating solutions. But almost everybody in the collaboration who we interviewed held a firm belief that the Grid will work; maybe not perfectly, but it will work.

A significant source of their confidence thus resides in a belief in the individual skill, competence and pragmatic creativity of physicists, as well as high-energy physics' formative context of collaboration. While GridPP employs people from other fields, the majority come from this 'elite science' (Traweek, 1988) which is highly competitive to enter. When asked about the likely success of LCG, a technical coordinator boils it down to cleverness:

> . . . because we are very clever people, we have a very clear and determined goal, we will make it work'.

Another source of confidence resides in the community's long history of success in computing. CERN for example accepted the problems of working with pre-production supercomputers from the days of the CDC 6600 through to the CRAY X-MP (Jones, 2004). Later they pioneered work on the Web (Berners-Lee & Fischetti, 1997), shifted early to use Open-source (Linux) server-farms, all driven by the need to do physics. Grid computing, it seems, is just another minor computing waypoint on the route to the truth about the universe. Equally importantly, the particle physics community enjoys an organizational culture which appreciates 'the aesthetic of imperfection' (Weick, 1999) and accepts failed attempts as part of a bigger process, fostering the confidence of individual innovators.

Organizing GridPP

Planned agility (Tables 1 & 7) refers to planning to improvise and preparing for change. In GridPP, it is recognized that *ad hoc* practices have to be supported by some financial planning, risk management, project milestones and resource allocation mechanisms. For this reason, extensive Gantt charts and schedules are produced, often in a preparation for research funding council reviews, but also serving as a minimal structure for the project. While a project manager was only appointed on the insistence of an IT industry representative sitting on the oversight committee, and the PMB finally settled on appointing a particle physicist (and 'friend' of GridPP) to the post, this role is now accepted as crucial to keeping the project on track. This is not however to say that the PM role focuses on traditional project management. Considering GridPP as in its essence 'experimental' and undertaking 'green-field research', the PMB focuses on supporting and justifying change as at the core of their minimal planning process.

> We wanted to establish the fact that we had the right to change our deliverables. So we set up this project map and we set up the formality of change forms. So this was to formalise our freedom to change the project . . . yes, we had a set of milestones but you know, we had a mechanism to change them because we have to be responsive.

Although schedules are constantly in flux, the project seeks never to lose sight of where they are and where they are heading.

Table 7. Paradoxes of organizing in GridPP

Unfolding	Planned agility	Planning
– Adhocracy – Constant changes and adaptation – Exploration – Flux	Planning to improvise; preparing for changes	– Common goal/shared vision – Memorandum of Understanding – Deliverables – Milestones – Project map – Quarterly reports
Practices	**Structured chaos**	**Structure**
– Bottom-up approach – Competition – Democratic discussions – Natural selection of parallel technical solutions – Transparency	Providing minimal structure to support improvisation	– Charismatic leadership – Collateral structure – Limited hierarchical command or authoritative management

> . . . people are looking at the overall targets of where people are trying to get to, rather than monitoring people on a daily or weekly basis. So we're looking for overall trends more than very small time-based ones.

The project maps and schedules, change forms and quarterly reports are tools designed to achieve various paradoxical goals; to display rationalized order, to acquire legitimacy, to cope with changes and to support or legitimize spontaneity. They also provide impetus to carry the project forward, even if the plan is tentative and has to be made real through day-to-day sensemaking and actions. This proactive mode of management is combined with a reactive mode of daily troubleshooting:

> We do everything we can in terms of advanced planning, so we have a staggered programme of sites in migrating, things like this. But ultimately what dominates is when we have done something that has gone wrong, or something has broken, or something doesn't work in experiments, or something like this, and we have to try and solve that.

In other words, there is a plan to improvise, routinize processes to stimulate improvisation and observation of their own improvisational activities (Miner *et al*., 2001). As one of the technical coordinators described, with an extended metaphor,

> You need your head in the clouds to see the big picture, but you very much need your feet on the ground because you have to put one foot in front of the other, and day to day we keep putting one foot in front of the other. . . .

Structured chaos (Tables 1 & 7) means providing a minimal structure to support improvisation. GridPP is a collaboration of institutes who work together under a memorandum of understanding. Management in GridPP does not rely on vertical lines of command, and while there is an extensive structure of management boards, committees and technical groups, they serve more as communication channels than hierarchies of authority. Managerial roles in the collaboration serve most of the time as representatives, spokesperson or coordinating facilitators, and when decisions (e.g. financial planning) have to be made centrally at the PMB, such decisions are open to scrutiny by the full collaboration. Most importantly, there is enormous respect to the technical knowledge at the grass root level. As one previous group leader stated:

> There's no strict hierarchy [. . .] the group leader doesn't get to say what to do. . . . We recognise it's the younger people that are much smarter and they're going to be making the papers . . . So it's kind of a federation, club . . . of smart academics who all want to do it and everyone trusts each other to be doing the best they can for the experiment. And that fundamental trust drives our particle physics group.

Different solutions often compete with each other within the collaboration for a while until one of them wins by forming more alliances or others die in a natural course, e.g. due to technical failures, low uptake, lack of funding or other circumstances. The technical systems then emerge from 'contests of unfolding' (Knorr-Cetina, 1999):

> The cream comes to the top. Things that work win out and that's how we worked it. (. . .) Nobody knew what the right approach was so you try several approaches and some win, some lose.

The 'natural selection' of technical solutions, as described by members of GridPP, allows elements of the Grid to emerge from dispersed and localized practices without an arbitrary or centrally imposed decision-making process. Although the middleware is developed by a European Grid development project (EGEE) centrally coordinated at CERN, it is modularized and each of the components is prototyped, released, deployed, tested and improved in an evolutionary manner. Beyond this core software there are often parallel technical solutions found in the project, such as some components of the middleware, or other software packages developed locally to help deploy, monitor or manage aspects of the Grid. The Grid environment thus consists of a mixture of 'ecosystems', in which multiple technical solutions coexist and even compete. Political influence and vested interests are reflected in such competition, but do not dictate outcomes. This is not to say that politics does not exist, but it is dispersed and mediated, and the influence of powerful actors is often dissipated, or contingent on sound technical judgment. As an interviewee commented

> Nobody, no matter, even if they were the most politically powerful person in EGEE, can force a broken piece of software to be deployed, because they will lose their political influence if they do that.

Learning to perform

Learned improvisation (Tables 1 & 8) refers to drawing upon past experience to cope with uncertainties and complexity of the present. The need to improvise in LCG stems from the innovative and exploratory nature of the task: the process has to be trial-and-error since nobody knows what exactly the end product will look like or what issues will emerge along the way.

Table 8. Paradoxes of learning in GridPP

Environment/present	Learned improvisation	History/culture
– Complexity of the Grid (multiple Grids, multiple groups of users) – Technological uncertainties – Time constraints (pressure of speed)	Drawing upon past experience to handle uncertainties and complexities of the present	– Pragmatic approach – Computing expertise/successes in PP – Tradition of distributed collaboration in experiments
Spontaneity	**Reflective spontaneity**	**Reflexivity/learning**
– Agility – Fast, incremental changes – Short cycle development – Trial-and-error	Recovering meaning from actions retrospectively	– Active informal face-to-face communication – Mailing lists, blogs, wiki, – Frequent multiple meetings, on site and virtual – Testing and monitoring

Moreover, the complexity, scale and distribution of the project means no one person can have a clear idea of the whole system (Hutchins, 1991); requirements cannot be pre-specified in detail; architectures are conjectures, and even the one centrally designed piece of technology, the middleware, has to be modularized and released gradually rather than in a big-bang manner.

Reliance on externally produced hardware and software also creates challenges by exposing external technological perturbations. Relying on the EGEE to provide the middleware, GridPP face an ongoing process of learning and adapting to immature software, and making it work at each individual site. For example, an undocumented change in the firmware of a set of hard disks included an error that had significant repercussions for GridPP as they struggled to isolate this irregular error among terabytes of distributed storage. Similarly, the release of a new version of the Scientific Linux operating system (on which LCG runs) created demands from some computer centres to upgrade GridPP to this new version particularly where computing resources were shared with other disciplines. Yet EGEE's software only ran on an even earlier version. Further issues occurred when some centres purchased 64-bit rather than 32-bit systems, requiring two different distributions of the software.

The response to this of those involved is not to control, predict or formalize, but rather to respond pragmatically and creatively at the time, drawing on the down-to-earth and creative approaches embedded in particle physics tradition (Lewis, 2000). As Dahlbom & Mathiassen (1993) describe, developers 'have to interact with the environment, accept the openness of the problem and the system to be developed, take into account the preferences and beliefs of problem owners and users, deal with the economical and political climate of the project and keep in step with the changes in the kind of technologies on which the project is dependent'. Developers should be 'scientific investigators' rather than 'economic agents' (ibid.). Indeed, particle physicists clearly bring their identity as 'scientific investigators' into computing.

> I think the people who come from a physics background are ultimately more pragmatic in computing. They see the computing as a tool to get a job done. And if it requires you to wrap sellotape around it to get it to work, then they will wrap sellotape around it . . . the physicists are happier with an ad hoc solution just to get the job done and push them through.

One of the resources that GridPP draws upon is their identity as physicists, and as noted the collaboration is designed as a physics experiment. The tradition of large-scale globally distributed collaborations (the ATLAS experiment, one of four at the LHC, has over 2000 members) and working on a distributed basis is well established and provides a solid basis for improvisation in the Grid development project. Such collaborations include students, technicians, engineers and physicists. Yet they have learnt at project management level how to organize collaborations to be pragmatic and drive towards solutions. In other words, the ability to improvise is the result of years of experience and learning. Such improvisation itself constitutes a further process of exploration and reflection which feeds into the organizational capability to improvise.

Reflexive spontaneity (Tables 1 & 8) indicates recovering meaning from improvisation retrospectively. The seemingly spontaneous practices at the low level are balanced by a level of reflexivity maintained by continuous and extensive communication flows. Particle physics collaborations are managed by what Knorr-Cetina (1999) refers to as ‘a fine grid of discourse’, channelling individual knowledge into the collaboration and providing it with a sort of ‘distributed cognition’. This web of communication includes a complex structure of boards, committees, and working groups which regularly hold meetings including online virtual meeting. For example, the PMB meeting takes place online every Monday where they discuss the status of the project and make action plans. The deployment team meets online on Tuesdays where they discuss technical issues. There are many other meetings taking place virtually or face-to-face during the week. Wikis, web pages and blogs are consultation points during the meetings. More importantly, members of GridPP subscribe to various mailing lists that carry constant exchanges of up-to-date information on problems and emerging solutions.

Such extensive communications embody both mutual monitoring and proactive sensemaking. It lies within the monitoring, accounting and making sense of the behaviour and performance of the system. Targets of service levels and regular data transfer exercises test the reliability and robustness of the systems hardware and software. Much GridPP discussions in meetings revolve around the results of such tests and monitoring statistics. Interpreting the statistics is not straightforward or free of controversy. One often hears remarks like ‘we have to understand what is causing this phenomenon’ or ‘find out what is behind the data’. In other words, retrospective sensemaking is an inherent and natural component in their process of system development. There is a ‘humming’ of the collaboration, talking ‘with itself, about itself’ (Knorr-Cetina, 1999), which maintains a constant collective reflexivity, as ‘the monitored character of the ongoing flow of social life’ (Giddens, 1984).

DISCUSSIONS AND IMPLICATIONS

We examine above the characteristics of the collaborative performance of GridPP that enable them to achieve distributed and scaled agility. Improvisation paradoxes have been used to make sense of the way that the Grid is developed. In this section, we reflect on the case material, draw implications for the wider discourse of agile systems development and provide some suggestions for those engaged in other distributed systems developments.

With multiple objectives and system development rationales in the community, the construction of Grid technology is a constant engagement and negotiation between a structured process and amethodical practices (Truex *et al*., 2000). Long-term goals, shared aims, preset deliverables, regular monitoring and proactive political legitimization are entangled with an 'unfolding ontology' (Knorr-Cetina, 1999), elements of which include: pragmatic outlook, fragmented and *ad hoc* practices, bricolage and improvised solutions, *post hoc* rationalization, as well as contested interests, internal competitions and democratic decision-making – 'a dialectic of resistance and accommodation' (Benson, 1977; Pickering, 1995).

The particle physicists, while not strictly following any pre-defined agile methods are aware of the challenges they face and have made deliberate and substantial effort to achieve a suitable development process. In other words, the agility seen here is not just an unintended consequence of loose coupling, a culture of improvisation and bricolage, intelligence, trust and pragmatism. Rather, it is a performance by knowledgeable actors who draw upon and enact certain properties of the distributed collaboration, such as minimal structure, flexible planning, extensive communication and social bonding, all serving to generate coherence, facilitate mutual understanding, promote sensemaking and to coordinate distributed work. The agency and knowledgeability of members of the project are central in this process. While no one serves as the mastermind of the project, the interaction and coordination among them give rise to a 'collective mindfulness' (Carlo *et al*., 2004) with 'a rich awareness of discriminatory detail and a capacity for action' (Weick *et al*., 1999). It takes real effort to maintain this collective mindfulness, without which distributed agility would not be possible or sustainable. Therefore, while agility can be described as an emergent property of the distributed collaboration, such emergence is very much enacted, involving degrees of deliberation and reflection, and instantiated in day-to-day practices.

What implications should we draw from our analysis of this case of 'collective agility'? What is presented may not be an ideal form of distributed agile systems development – after all, it is not in a commercial environment or facing immediate safety critical concerns such as in health care or air traffic control. Yet there is a lot we can learn. From the perspective of organizational performance, collective agility is about accepting what is unpredictable and uncontrollable, while actively enacting those organizational dimensions that generate capabilities to perform under such circumstances. Table 9 presents examples of organizational practices from the case that could be useful to practitioners who share the interest in what it takes to 'be agile'. In the sections below, we explore further by asking the questions of *when*, *what*, *who*, *where* and *how* is such case-specific collective agility *performed*. Our implications should be taken in the round – we see each as part of a cumulative recommendation for those engaged in similar practices rather than an isolated concept.

When is collective agility performed?

Agility is indicated when faced with environmental turbulences, uncertainties and an innovative or exploratory task, as is the case with GridPP. Yet organizational improvisations come with risks, and for example, may not be the most efficient or effective way to tackle certain problems,

Table 9. Key organizational practices in GridPP

- Draw upon past experience to handle new tasks;
- Continuous reflection and learning;
- Extensive communications within and between different groups, with an emphasis on face-to-face informal communication;
- Work with power users; cultivate user communities;
- Project leader articulates clear vision and shared goals;
- Use high level milestones and deliverables to create momentum, but be ready to change them;
- Share knowledge by mailing lists, wiki, blogs, etc;
- Cultivate community bonding and shared identity;
- Develop trust, loyalty and mutual support;
- Motivate and rely on good people;
- Maintain high level of transparency within the project;
- Allow mistakes and unsuccessful explorations;
- Allow parallel solutions to compete with each other when resources permit; it might be a faster and safer way of achieving a goal.

despite being preferred by a community drawing on their past success. The particle physics community's tradition of experimental scientific investigation and pragmatic problem solving means an agile approach is 'natural' rather than contrived, yet this may also means it is 'assumed' rather than 'considered'. Over-reliance on improvisation can also lead to an amplification of unexpected events and crises, self-generating a negative spiral of uncertainties and complexities (Cunha *et al*., 1999). Communities engaging in large scale and distributed systems development are thus faced with the challenge of getting the appropriate mixture of improvisation and structure. As expressed by the concept of *learned improvisation*, we argue agility can (to degrees) be learnt. Communities accustomed to more formal management approaches are not incapable of achieving agile performance. On the contrary, our research suggests, organizations with established routines and strong cultures to draw upon might be better equipped to improvise than those without. But this needs cultivation of the space and motivation to diverge from or reinterpret established routines. Collective agility is performed when some 'tools are dropped' (Weick, 1993a), and surprise, risk and wonder are accepted in the community.

What is (the spirit of) collective agility?

We argue here that collective agility is supported by a sustained mood of *anxious confidence*. Anxiety stems from the nature of innovative tasks, and in the sense of urgency, pressure and demand for speed in problem solving. A successful innovative community needs a countervailing level of confidence which can stem from strong individual skills and experience under demanding conditions, as well as a history of technical success as well as appropriate social settings. This confidence can arise in part from an appreciation for the 'aesthetics of imperfection'. An atmosphere of experimentation, trust, shared goal and emotional bonds provides individuals and groups with confidence to make mistakes, in the knowledge that failures are legitimate and can contribute to the cause of the community.

Who undertakes collective agility?

The competence of the people, the level of determination and motivation and how well the group gets on were identified by GridPP members as the most positive aspects of the project. Performing collective agility poses a high demand on individual skills and mental attitudes. Like most professional domains, recruits are expected to be self-motivated, good communicators and able to work in a collaborative environment. Distinctively though, GridPP prefers people who are familiar with the institutional culture of the particle physics community, and who are thus motivated to step up and do the dirty work when necessary without explicit instruction or reward. The level of commitment, devotion and voluntarism appear higher than one might observe in some commercial contexts. While individuals certainly have personal career interests at stake, many express a sense of pride in working for a higher cause, perhaps explaining their willingness to undertake unpopular tasks when needed.

Where does collective agility happen?

The literature suggests that both improvisation and agility are more easily performed in small groups, such as a jazz ensemble or small development teams. Our case shows agility is possible in a large and distributed group, when the ‘ambience’ is right, although achieving this is itself a major challenge (see also Ramesh *et al*., 2006). Community bonds can alleviate many difficulties but require effort to maintain. Even though GridPP members are accustomed to virtual meetings and a large number of emails, they still emphasize the importance of face-to-face communication and travel extensively to meet up. Being reliant on delivery from many remote partners without the authority over them is often a source of frustration, thus the ability to exert gentle pressure, to persuade and to negotiate are important elements in coordinating a collective performance. Meanwhile, barriers of communication or an overload of information can also create inefficiencies in a non-hierarchical community.

How is collective agility performed?

As has been repeated throughout the paper, agility requires a mental attitude to, in the words of one interviewee, ‘let go of control’, yet this does not mean anarchy. High-level planning and a minimal structure are required. For GridPP, this consists of alignment with goals of doing new physics, a clear orientation towards the LHC objectives, a shared culture among participants and a carefully crafted minimal structure of project management, and communication channels to allow local ‘clusters of expertise’ to interact. Improvisation at the local level is complimented by structuring at the distributed level to maintain cohesiveness across the project and to create a sense of community among the independent-thinking actors. Finally organizational improvisation comes with risks. The lack of formal planning and reflexivity may mean that exploitation of novel ideas and knowledge is limited despite a great deal of exploration, thus creating ‘opportunity traps’.

CONCLUSIONS

This paper considers agile system development practice from the perspective of organizational performance, reflecting an understanding that systems development processes and activities cannot be discussed in a vacuum but must be considered in terms of how, in given contexts, they become embodied within a set of roles, attitudes and working practices adopted by people – as a *performance*. From this case study, we observe that the LHC-Grid unfolds in a constant negotiation and mediation between design and bricolage (‘working things out’), between planning and improvisation, and between enough success and tolerable and instructive failure. Drawing on previous work on paradox, we use a set of improvisation paradoxes as a framework to examine system development practices within this distributed development context. This framework, and the attention to collective performance, enables us to elaborate and explore elements often pushed to the background in discussions of system development, such as environmental conditions, individual skills, professional cultures, organizational structures, communication patterns and interpersonal relationships. The case study demonstrates in this community of science known to be most rationalistic and analytical, systems development is actually more like an ‘art’ – visionary, experiential, passionate, agile and emergent.

This study has limitations and thus suggests some future research directions. First the context of GridPP is clearly distinctive. Experimental physics offers an environment that in many ways embraces an agile approach. Studies in other contexts that explore similar themes would be useful, for example in the various projects around the world that are developing national health care information infrastructures (Coiera, 2009), or those developing systems for the cloud (Buyya *et al*., 2009). This work might also suggest some comparison with the ways in which the internet and its core systems and services have developed using a ‘community centric’ development model, as well as some aspects of the open source model (Tuomi, 2002; 2005). While the context and outcomes in such domains are rather different, the performative analysis seems to have strong resonance. We also acknowledge that the period of this study was one of development and testing rather than of operations and the phase change may have significant consequence on the approach to system development. Indeed, towards the end of the study reported here, there were signs that could indicate a greater emphasis on a more structured approach within the LCG.

Despite the limitations discussed above, the contributions from this research are, we believe, significant.

First we provide a conceptualization of agility (collective agility) that differs substantially from the more common concerns with agile methods and behaviour, or with an organization’s agile potential or capabilities. In contrast, we adopt a performative ontology and develop an understanding of agility through the concept of enacted emergence. Agility for us is an expression of what people do or achieve, rather than what they might do or capabilities they hold. We argue that collective agility, as an organizational performance, emerges from collective enactment of certain qualities and processes in the distributed community.

Second, we derive six improvisation paradoxes from the literature and use them to give a stronger conceptualization to the work of GridPP members as they negotiate the contradictory pressures for order and innovation. These paradoxes are used to underline the dynamics of the agile performance; as an expression of, and (to a degree) the resolution of, fundamental tensions. It is the mutual constitution of elements in tension that allow agility to emerge.

Third, we offer a contribution to practice by drawing implications from the case, presented as the when, what, who, where and how of collective agility, and covering both useful practices identified and risks to be aware of. These recommendations highlight the means by which collective agility might be achieved and maintained, and offer insights for other domains attempting to construct large-scaled distributed infrastructure in an agile fashion. For example, Ramesh *et al*. (2006) assert that distributed agility faces the challenge of communication, lack of control and lack of trust. Our study suggests that these may not be causes but symptoms of a broader failing to understand the nature of agility within such a context. Thus to managers who want to achieve some of the attributes of a collective agile performance, we suggest that communication, control and trust cannot be isolated from more complex and comprehensive efforts to support and cultivate an innovative culture within the distributed community, and require reflection on questions of balance among the various paradoxical tensions embedded.

Finally the ‘enacted emergence’ of collective agility highlights the need for ongoing performances – agility is not a ‘per-project’ or even less, ‘per-phase’ activity and cannot be achieved by a top-down ‘change programme’. Rather it is a performance that is reflective of multiple collective organizational practices. Hence, achieving it must be a long-term aspiration requiring attention and adjustments over time, and like other institutional practices, collective agility may be fragile and easily broken. For LCG it might be that the future would be different, and that contractual relationships and a reliance on technical and managerial rationality would prevail once the LHC data begins to flow in bulk. What is clear is that many domains where large-scale distributed systems are under development can learn from this case and the paradoxical nature of collective agility.

REFERENCES

Abrahamsson, P., Conboy, K. & Wang, X. (2009) Editorial: ‘Lots done, more to do’: the current state of agile systems development research. *European Journal of Information Systems*, **18**, 281–284.

Ackerman, M. & Halverson, C. (1998) Considering an Organization’s Memory. Proceedings of the ACM 1998 Conference on Computer Supported Cooperative Work, Seattle, WA, 39–48.

Avgerou, C. & Cornford, T. (1993) A review of the methodologies movement. *Journal of Information Technology*, **5**, 277–286.

Bansler, J. & Bodker, K. (1993) A reappraisal of structured analysis: design in an organisational context. *ACM transactions on Information Systems*, **11**, 165–193.

Bansler, J.P. & Havn, E.C. (2004) Improvisation in information systems development. In: *Information Systems Research: Relevant Theory and Informed Practice (IFIP 8.2)*, Kaplan, B., Truex, D.P., Wastell, D., Wood-Harper, A.T. & DeGross, J.I. (eds.), pp. 631–646. Kluwer Academic Publishers, Manchester, UK.

Barrett, F.J. (1998) Coda: creativity and improvisation in jazz and organisations: implications for organisational learning. *Organization Science*, **9**, 605–622.

Baskerville, R.L. (2006) Artful planning. *European Journal of Information Systems*, **15**, 113–115.

Baskerville, R. & Pries-Heje, J. (2004) Short cycle time systems development information systems journal. **14**, 237–264.

Baskerville, R., Travis, J. & Truex, D. (1992) Systems without method: the impact of new technologies on information systems projects. In: *The Impact of Computer Supported Technologies on Information Systems Development*, Kendall, K.E., Lyytinen, K. & DeGross, J.I. (eds), pp. 241–269. North-Holland, Amsterdam, The Netherlands.

Baskerville, R., Ramesh, B., Pries-Heje, J. & Levine, L. (2006) High-speed software development practices: what works, what doesn't. *IEEE IT Professional*, **8**, 29–36.

Beck, K. & Andres, C. (2005) *Extreme Programming Explained*. Addison-Wesley, Upper Saddle River, NJ, USA.

Benson, J.K. (1977) Organisations: a dialectical view. *Administrative Science Quarterly*, **22**, 1–21.

Berliner, P.F. (1994) *Thinking in Jazz: The Infinite Art of Improvisation*. The University of Chicago Press, Chicago.

Berners-Lee, T. & Fischetti, M. (1997) *Weaving the Web*. Harper, San Francisco, CA, USA.

Boehm, B. (1988) A spiral model of software development and enhancement. *Computer*, **21**, 61–72.

Britton, D., Clark, P., Coles, J., Colling, D., Doyle, A., Fisher, S.M. *et al*. (2004) *A Grid for Particle Physics – From Testbed to Production*. GridPP, Proceedings of the UK e-Science All Hands Conference, Nottingham, September 2004.

Brooks, F. (1979) *The Mythical Man-Month*. Addison-Wesley, New York, USA.

Brown, J.S. & Duguid, P. (1991) Organisational learning and communities-of-practice: toward a unified view of working, learning, and innovation. *Organization Science*, **2**, 40–57.

Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J. & Brandic, I. (2009) Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, **25**, 599–616.

Carlo, J.L., Lyytinen, K. & Boland, R.J. (2004) Systemic risk, IT artifacts, and high reliability organisations: a case of constructing a radical architecture. *Sprouts: Working Papers on Information Environments, Systems and Organisations*, **4**, http://sprouts.aisnet.org/4-4

Chae, B. & Bloodgood, J.M. (2006) The paradoxes of knowledge management: an eastern philosophical perspective. *Information and Organisation*, **16**, 1–26.

Chae, B. & Poole, M.S. (2005) Surface of emergence in systems development. *European Journal of Information Systems*, **14**, 19–36.

Ciborra, C. (1996) The platform organization: Recombining strategies, structures, and surprises. *Organization science*, **7**, 103–118.

Ciborra, C. (1999) A theory of information systems based on improvisation. In: *Rethinking Management Information Systems*, Currie, W. & Galliers, B. (eds), pp. 136–155. Oxford University Press, Oxford, UK.

Ciborra, C.U. (2001) In the mood for knowledge: a new study of improvisation. Social Study of Information Technology Workshop, London School of Economics and Political Science, London, UK.

Ciborra, C. (2002) *The Labyrinths of Information*. Oxford University Press, Chicago.

Ciborra, C., Braa, K., Cordella, A., Dahlbom, B., Failla, A., Hanseth, O. *et al*. (2000) *From Control to Drift: The Dynamics of Global Information Infrastructures*. Oxford University Press, Oxford.

Cohen, M.D. *et al*. (1972) A garbage can model of organisational choice. *Administrative Science Quarterly*, **17**, 1–25.

Coiera, E. (2009) Building a national health IT system from the middle out. *Journal of American Medical Information Association*, **16**, 271–273.

Conboy, K. & Fitzgerald, B. (2004) Toward a conceptual framework of agile methods: a study of agility in different disciplines. *Proceedings of the WISER'04*, Newport Beach, CA, USA.

Corbin, J. & Strauss, A. (2008) *Basics of Qualitative Research*. Sage, Los Angeles, CA, USA.

Crossan, M.M. (1998) Improvisation in action. *Organization Science*, **9**, 593–599.

Crossan, M.M., Lane, H.W., White, R.E. & Klus, L. (1996) The improvising organization: where planning meets opportunity. *Organisation Dynamics*, **24**, 20–35.

Cunha, M.P., Cunha, J.V. & Kamoche, K. (1999) Organisational improvisation: what, when, how and why. *International Journal of Management Reviews*, **1**, 299–341.

Curtis, B., Krasner, H. & Iscoe, N. (1988) A field study of the software design process for large systems. *Communications of the ACM*, **31**, 1268–1287.

Dahlbom, B. & Mathiassen, L. (1993) *Computers in Context: The Philosophy and Practice of Systems Design*. Malden, MA, USA.

DeMarco, T. (1978) *Structured Analysis and Systems Specification Yourdon*. Prentice-Hall Software, New York, USA.

DeRemer, F. & Kron, H. (1975) Programming-in-the large versus programming-in-the-small. *Proceedings of the International Conference on Reliable Software*, pp. 114–121, ACM, Los Angeles, CA, USA.

Dyba, T. (2000) Improvisation in small software organisations. *IEEE Software*, **17**, 82–87.

Flor, N.V. (2006) Globally distributed software development and pair programming. *Communication of ACM*, **49**, 57–58.

Foster, I., Kesselman, C. & Tuecke, S. (2001) The anatomy of the grid: enabling scalable virtual organisations. *The International Journal of High Performance Computing Applications*, **15**, 200–222.

Fowler, M. & Highsmith, J. (2001) The Agile Manifesto. Software Development (August).

Giddens, A. (1984) *The Constitution of Society: Outline of the Theory of Structuration*. Polity Press, Cambridge, UK.

Hatch, M.J. (1999) Exploring the empty spaces of organising: how improvisational jazz helps redescribe organisational structure. *Organization Studies*, **20**, 75–100.

Highsmith, J. (2002) *Agile Software Development Ecosystems*, Addison Wesley Professional.

Hutchins, E. (1991) Organising work by adaptation. *Organization Science*, **2**, 14–39.

Hutchins, E. (1995) *Cognition in the Wild*. The MIT Press, Cambridge, MA, USA.

Jones, C. (2004) Computing at CERN: the mainframe era. *CERN Courier*, **44**, 32–35.

Kamoche, K., Cunha, M.P.E. & Vieira da Cunha, J. (2003) Towards a theory of organisational improvisation: looking beyond the jazz metaphor. *Journal of Management Studies*, **40**, 2023–2051.

Kautz, K., Madsen, S. & Nørbjerg, J. (2007) Persistent problems and practices in information systems development. *Information Systems Journal*, **17**, 217–239.

Knorr-Cetina, K. (1999) *Epistemic Cultures: How the Sciences Make Knowledge*. Harvard University Press, Cambridge, MA, USA.

Lanzara, G.F. (1999) Between transient constructs and persistent structures: designing systems in action. *Journal of Strategic Information Systems*, **8**, 331–349.

LaPorte, T.R. (1996) High Reliability Organizations: Unlikely, Demanding and At Risk. *Journal of Contingencies and Crisis Management*, **4**, 60–71.

Lee, O.-K., Banerjee, P., Lim, K.H., Kumar, K., Hillegersberg, J.V. & Wei, K.K. (2006) Aligning IT components to achieve agility in globally distributed system development. *Communications of the ACM*, **49**, 49–54.

Lee, G., DeLone, W. & Espinosa, J.A. (2006) Ambidextrous coping strategies in globally distributed software development projects. *Communications of the ACM*, **49**, 35–40.

Lewis, M. (2000) Exploring paradox: toward a more comprehensive guide. *Academy of Management Review*, **25**, 760–776.

Mathiassen, L. & Pries-Heje, J. (2006) Business agility and diffusion of information technology. *European Journal of Information Systems*, **15**, 116–119.

Mathiassen, L. & Vainio, A.M. (2007) Dynamic capabilities in small software firms: a sense-and-response approach. *IEEE Transaction of Engineering Management*, **54**, 522–538.

Miner, A.S., Bassoff, P. & Moorman C. (2001) Organisational improvisation and learning: a field study. *Administrative Science Quarterly*, **46**, 304–337.

Mintzberg, H. (1995) The structuring of organisations. In: *The Strategy Process: European Edition*, Mintzberg, H., Quinn, J.B. & Ghoshal, S. (eds), pp. 350–371. Prentice Hall Europe, Hertfordshire, UK.

Mintzberg, H. & McHugh, A. (1985) Strategy formation in an adhocracy. *Administrative Science Quarterly*, **30**, 160–197.

Mirvis, P.H. (1998) Variations on a theme: practice improvisation. *Organization Science*, **9**, 586–592.

Moorman, C. & Miner, A.S. (1998) The convergence of planning and execution: improvisation in new product development. *Journal of Marketing*, **62**, 1–20.

Nandhakumar, J. & Avison, D.E. (1999) The fiction of methodological development: a field study of information systems development. *Information Technology & People*, **12**, 176–191.

Orlikowski, W. (1996) Improvising organisational transformation over time: a situated change perspective. *Information Systems Research*, **7**, 63–92.

Orlikowski, W.J. (2000) Using technology and constituting structures: a practice lens for studying technology in organisations. *Organization Science*, **11**, 404–428.

Overby, E., Bharadwaj, A. & Sambamurthy, V. (2006) Enterprise agility and the enabling role of information technology. *European Journal of Information Systems*, **15**, 120–131.

Parnas, D. & Clements, P. (1986) A rational design process: how and why to fake it. *IEEE Transaction of Software Engineering*, **12**, 346–357.

Pickering, A. (1995) *The Mangle of Practice: Time, Agency and Science*. University of Chicago Press, Chicago.

Poole, M.S. & Van de Ven, A.H. (1989) Using paradox to build management and organisation theories. *The Academy of Management Review*, **14**, 562–578.

Ramesh, B., Cao, L., Mohan, K. & Xu, P. (2006) Can distributed software development be agile. *Communications of the ACM*, **49**, 41–46.

Reifer, D.J., Maurer, F. & Erdogmus, H. (2003) Scaling agile methods. *IEEE Software*, **20**, 12–14.

Sambamurthy, V., Bharadwaj, A. & Grover, V. (2003) Shaping agility through digital options: reconceptualizing the role of information technology in contemporary firms. *MIS Quarterly*, **27**, 237–263.

Smith, W.K. & Tushman, M.L. (2005) Managing strategic contradictions: a top management model for managing innovation streams. *Organization Science*, **16**, 522–536.

Traweek, S. (1988) *Beamtimes and Lifetimes: The World of High Energy Physicists*. Harvard University Press, London, UK.

Truex, D.P. & Baskerville, R. (1998) Deep structure or emergence theory: contrasting theoretical foundations for information systems development. *Information Systems Journal*, **8**, 99–118.

Truex, D., Baskerville, R. & Travis, J. (2000) Amethodical systems development: the deferred meaning of systems development methods. *Accounting, Management and Information Technology*, **10**, 53–79.

Tuomi, I. (2002) *Networks of Innovation*. Oxford University Press, Oxford, UK.

Tuomi, I. (2005) The future of open source. In: *How Open Is the Future?*, Wynants, M. & Cornelis, J. (eds), pp. 429–459. VUB Brussels University Press, Brussels, Belgium.

Venters, W. & Cornford, T. (2006) Introducing Pegasus: an ethnographic research project studying the use of grid technologies by the UK Particle Physics Community. *Proceedings of the National e-Science Center Conference*, Manchester, UK.

Wade, M. & Hulland, J. (2004) The Resource based view and information systems research: review, extension and suggestions for future research. *MIS Quarterly*, **28**, 107–138.

Weick, K.E. (1993a) The collapse of sensemaking in organisations: the Mann Gulch disaster. *Administrative Science Quarterly*, **38**, 628–652.

Weick, K.E. (1993b) Organisational redesign as improvisation. In: *Organisational Change and Redesign: Ideas and Insights for Improving Performance*, Huber, G.P. & Glick, W.H. (eds), pp. 346–379. Oxford University Press, New York, USA.

Weick, K.E. (1998) Improvisation as a mindset for organisational analysis. *Organization Science*, **9**, 543–555.

Weick, K.E. (1999) The aesthetic of imperfection in orchestras and organisations. In: *Readings in Organisation Science*, Cunha, M.P. & Marques, C.A. (eds), ISPA, Lisbon, Portugal.

Weick, K.E. (2001) Enactment processes in organisations. In: *Making Sense of the Organization*, Weick, K.E. (ed.), pp. 179–206. Blackwell, Malden, MA, USA.

Weick, K.E. & Roberts, K.H. (1993) Collective mind in organisations: heedful interrelating on flight decks. *Administrative Science Quarterly*, **38**, 357–381.

Weick, K.E., Sutcliffe, K.M. & Obstfeld, D. (1999) Organising and the process of sensemaking. *Organization Science*, **16**, 409–421.

Williams, L. & Cockburn, A. (2003) Agile software development: it's about feedback and change. *IEEE Computer*, **36**, 39–43.

Biographies

Yingqin Zheng is Senior Lecturer at the Center for Computing and Social Responsibility, De Montfort University. Her research interests lie with organizational and social dynamics of information systems in developed and developing contexts. Her work encompasses collaborative distributed innovations in a broad sense, and theories and practices in relation to the role of information and communication technologies for human development.

Will Venters is a Lecturer in Information Systems at the London School of Economics. His research focuses on the distributed work practices associated with systems development and implementation. In particular he is interested in collaborative and inter-organizational ecologies to develop information systems, and how systems are socially constructed during development and use.

Tony Cornford is Senior Lecturer in Information Systems at the London School of Economics. His research interests are in information systems implementation (broadly understood), in particular in the areas of science, healthcare and medicine. This interest incorporates issues of systems development models and practices, evaluation of information systems and information technologies, and the study of systems in use and users contributions to their performance.