Edinburgh Research Explorer

Post local forms of repair

Post Local Forms of Repair:

The (Extended) Situation of Virtualised Technical Support

Neil Pollock[a],[*], Robin Williams[b], Luciana D'Adderio[b] and Christine Grimm[c]

[a] School of Business, University of Edinburgh, William Robertson Building, George Square, Edinburgh, Midlothian EH8 9JY, UK
[b] Institute for the Study of Science, Technology and Innovation, University of Edinburgh, Old Surgeon's Hall, High School Yards, Edinburgh EH1 1LZ, UK
[c] SAP (Switzerland) Inc, SAP Research, CEC Zurich, Kreuzplatz 20, CH-8008 Zurich, Switzerland

Abstract

We address the seemingly implausible project of moving the technical support of complex organisational technologies online. We say 'implausible' because from the point of view of micro-sociological analysis and the influential work of Julian Orr (1996) there is a consensus that the diagnosis and resolution of technical failures is an intrinsically 'localised affair' (i.e., rooted within a specific place and time). Notwithstanding this view, technology producers have been pushing in the recent period to develop online forms of support. Today, and particularly in the area of organisational software, many technical failures are now repaired at a distance. How is this possible given the consensus amongst sociologists? Drawing on ethnographic fieldwork conducted at a major software producer we show how repair work has been recast and inserted in a new geographical and temporal regime. This has implications for how sociologists of technology conceptualise the *nature* and *practice* of technical failure but also the *time* and *situation* in which it occurs. We attempt to refocus understandings of technical problems from a preoccupation with their rootedness onto how they are lifted out of local contexts and passed around globally distributed offices in search of requisite specialist expertise. Importantly, whilst virtualisation appears a seemingly effective means to resolve failures it also has negative consequences. Whereas in more traditional types of technical support place-based social relations are seen to bear the burden of controlling and regulating support, in online forms other means have to be found. Our conceptual aim is to move away from a view of repair revolving exclusively around *the situation* conceived of as a 'small place'. Rather, since support work is increasingly 'stretched out' across a global network of labs connected up by technologies, it now takes place across an *extended situation*. We work up this notion first to highlight how aspects once seen as central to localist forms of analysis are no longer the only organising features as technical work moves online and second to demonstrate the various ways in which the locales for this work are now mediated by technology.

Keywords: Extended situation; situated; global; local; technical support; passing regime

* Corresponding author. Tel.: +44 (0) 131 6511489; fax: +44 (0) 131 6683053.

E-mail addresses: neil.pollock@ed.ac.uk, neil.pollock@edinburgh.ac.uk (N. Pollock).

1

1. Introduction[1]

Resolving technical failures is said to be a highly localised and situated affair. Julian Orr (1996) wrote one of the most comprehensive sociological studies of technical support discussing in rich detail the work of engineers repairing photocopiers and paving the way for studies of situated work practice – which we describe as 'Orr-type' studies. Drawing inspiration from the growing influence of contemporary interactionist and situated perspectives (notably Suchman [1987]), as well as an associated enthusiasm for ethnographic studies, Orr postulated that technical problems were *rooted* in a specific place and time and thus could not be resolved without a sufficient understanding of (and interaction with) that context (Barley 1996; Henke 1999). Localist views of repair and maintenance have gone on to become a highly important and productive form of analysis acting as a catalyst for more general studies of work practice (see for instance Bechky's [2006] discussion of the wide adoption of Orr's work within the field of Organisation Studies). Through their influence, however, these authors have arguably established a version of what Ophir and Shapin (1991) some years ago called the 'successor problem'. That is, in countering the more dominant discourses of globalisation in which the boundaries of space and time are seen to collapse in the face of technology driven rationalisation (Castells 2000; Cairncross 2001), or assumptions that technical work is amenable to straightforward codification and systemisation (Cowan & Foray 1997), it is now difficult to envisage repair as anything other than a highly local and entangled affair.

This view is increasingly found wanting, however, especially when confronted with the new forms of technical support afforded by information and communication technologies (ICTs), where, in some cases, technical problems appear no longer as strongly attached to their contexts as they once were. Our empirical setting is information systems (specifically, the repair of the large packaged enterprise-wide systems found in the majority of today's organisations). We might presume, given our reading of the sociological literature on technical support, that the work necessary to fix a failing enterprise system would be one of those occasions where, by necessity, repair would require Orr-type forms of localised support. These are highly complex technologies requiring both 'tailoring' (to fit organisational needs) and 'integration' (with existing information infrastructures).[2] Yet this is seemingly no longer the case. We conducted

2 One has only to read the vast (and still growing) academic and practitioner literature on the various difficulties adopters experience whilst attempting to implement these systems. These are highly complex and intricate packages that are built to work in a huge variety of settings, interface and run on different

research on a world leading global software vendor. Crucially, whilst it receives hundreds of thousands of requests for help every year, today, and, in stark contrast to the situation a few years ago, only a very small number of these now require localist forms of support. Instead, the bulk of users receive help remotely, through a virtualised medium, in which support staff rarely (if at all) meet with the users of failing systems, have little or no specific knowledge of their local site, and interact with them through a restricted channel.[3] Moreover, not only are problems seldom dealt with locally, but they are also resolved through the so-called 'follow the sun' approach (Aneesh 2006). That is, customers' failing systems are worked on remotely in one office and, when those experts go home for the night, the problems are passed on to other staff in a different time zone. This produces two important results: failures receive immediate and constant attention *wherever* and *whenever* they arise; but it also common for less tractable problems to have travelled the world (and been worked on by many different teams) before a satisfactory resolution is found.[4]

2. Research Aim

The general aim of this paper is to unpack this seeming contradiction. Given the current sociological understanding of the rooted and situated nature of technical problems, fully virtualised forms of technical support should *not* be possible. Clearly, however, there are differences between the technologies and forms of practice we are discussing and the ones theorised by scholars more than a decade ago now. Both technologies and practices have evolved. Orr dealt with mechanical repair whilst we are interested in software, which affords different possibilities. Of course, whilst some of the issues remain the same there are a number of new ones. However, more importantly, the bulk of discussions within what has become known as the Social Study of ICTs still view technology (often implicitly) through a localist lens (see for instance the discussion of the embedded nature of knowledge in Walsham [2001a] or Nicholson & Sahay [2004]). We make this link to show that these approaches are not well equipped to handle what we call post local forms of practice. Through the term 'post local', we refer to important shifts occurring in *how*, *when* and *where* technical problems are managed and resolved. The technical support of software has been recast and inserted into a new geographical and temporal regime, which has implications for how sociologists conceptualise the nature and

---

platforms and where there are ultimately an enormous number of permeations of these systems as they meet different socio-technical settings (Gable 2001; Hirt and Swanson 2001; Jansen et al. 2006). More analytically, we might describe these kinds of systems (and organisational information systems more generally) as exhibiting enduring complexity. The diversity of socio-technical arrangement – the different technical and organisational contexts in which they are implemented – means that they continue to resist straightforward black-boxing (Pollock & Williams 2009).

3 The vendor reports that whilst it receives in excess of 800,000 calls for help each year, today, only 500 or so of these require a support specialist to actually to travel out to visit a site (vendor presentation). The vast majority of complex technical failures are managed, diagnosed and resolved remotely.

4 We do not think our case is unusual – many of the larger software providers have moved, or are attempting to move, technical support online (see for instance Orlikowski [1996] who discussed an early attempt by one firm to automate and formalise the repair of software).

practice of repair as well as the space and time in which it occurs. Arguably, this and earlier forms of scholarship on which this paper draws (specifically Knorr Cetina & Bruegger [2002]) have implications also for the understanding of ICT mediated work more generally.

More specifically the paper has three empirical aims as well as a wider set of conceptual ambitions. The first is to show how the nature and practice of this form of repair is changing. Technical problems are no longer simply rooted in local contexts but can be lifted out and passed on elsewhere to be dealt with in novel ways. The second is to show that, whilst virtualisation appears to be a practicable solution for many kinds of problem solving, it also has a series of unintended consequences (we highlight in particular the 'temporal' pressures support staff are under as well as the various strategies they adopt to ameliorate this). The third is that whereas in Orr-type situations place-based social relations bear the burden of controlling and regulating support, in post local forms other means have to be found. We discuss the emergence of a 'passing regime', which is how we analyse the various attempts to modify and regulate online interactions to eliminate or at least mitigate undesired consequences. The conceptual aim is to move away from a view of repair revolving exclusively around the situation – or at least the situation conceived of as a 'small place' (Harris 1998). Rather, since support work is increasingly 'stretched out' (Nicolini 2007) across a global network of labs connected up by technologies, it now takes place across a *globalised* or *extended* situation. We work up this notion to highlight how aspects once seen as central to localist analysis (co-location, proximity, etc) are no longer the primary organising features of technical support as others begin to emerge, but also to show how the 'where' of this form of work is now mediated by technology.

The material for the paper was gathered during ethnographic research in one of the vendor's 'labs' responsible for delivering technical support. We supplemented this with further research on a user organisation receiving remote help from the vendor. The paper is structured as follows. We first introduce a brief discussion of the sociological and situated accounts of technical support focusing primarily on the work of Julian Orr (1996) and Brian Pentland (1992) who offer contrasting viewpoints on the repair process. Before turning to our empirical material, we provide background information on the vendor's rationale for embarking on this project as well as detail on the specific shaping of the support technology (which both have a particular bearing on how the process is played out). We conclude by laying out our framework for characterising the changes in the structure and governance of repair through an on-line virtual medium.

3. Localised Views on Technical Support

3.1. Orr-type Studies

Much of the sociological literature views technical support through the lens of communitarian based expertise and situated forms of action. This is a rich body of work, which, for our purposes, can be distilled down into five aspects. The repair of a failing technology (1) demands not simply abstract or formal knowledge but also specific *understanding of the social context* in which the artefact is embedded. Orr's (1996) observations arose in the context of efforts by management to standardise the process of photocopier repair. Holding a rationalist viewpoint, managers attempted to separate formal knowledge about support from the actual practice of doing support but found documenting the various steps required to be difficult. Certain kinds of failure (and photocopiers) were 'peculiar', requiring expertise not easily captured in documented processes. In particular, (2) machine errors were marked by various forms of *local contingency*. Although Orr (1996) does not characterise it in these terms, it might be suggested that technical repair takes place primarily in an uncontained and 'open world' (Pentland 1995, 1997). Failures, when they occur, are initially 'ill-structured', meaning the boundaries around problems (what's causing it? where is it?) are yet to be established (ibid.). This can be compounded by the sheer number (and variety) of failures which specialists have to deal with. In this respect, technical support has been likened to the emergency treatment given by hospitals where the needs of those seeking help outstrip the resources of those providing it (and scholars have gone as far as borrowing some of the same terminology). Pentland (1995) suggests that in order to sort serious problems from trivial ones, features such as 'triage' are common to both worlds.

Added to this, and a further form of contingency, support specialists are seldom in command of the settings where failure occurs. They are forced to 'react' to all kinds of issues when visiting sites. As Orr writes:

> …technicians are responsible in a world in which they have very little control. Their job is to respond to trouble and to anticipate and avoid trouble when they can, but the setting in which they perform is largely constructed by other people, inhabited by other people and inherently unpredictable (1996: 158).

For the above reasons, Orr and others have discussed how (3) technical support cannot be understood without taking into account *the situation*. It is only through visiting settings, observing forms of use, and interacting with users that specialists can gain the largely tacit understandings crucial for repair (Barley 1996; Henke 1999). Proximity in particular is seen to facilitate the exchange of knowledge and foster close relationships not possible in other ways. This was seen most visibly in Orr's description of how (4) knowledge about problems was often

supplemented by *interactions with colleagues*. Seemingly 'being together' allowed technicians to find solutions to problems not obtainable in other ways. Orr writes: "[t]he practice of diagnosis is done through narrative, and both diagnosis and process are preserved and circulated amongst the technicians through war stories, anecdotes of their experience" (1996, 104). The circulation of knowledge about problems is an important aspect and we return to it below. Finally, (5) *the territorial space* in which users were located also figured in the resolution of problems. Orr described how technicians 'owned' particular regions and had 'tighter relationships' with those in their area as opposed to those 'outside' (even to the extent that this affected the levels of help provided). He writes: "[t]erritorial boundaries provide a reference for all action, whether one is inside one's territory or outside" (ibid. 64).

To summarise, these accounts have established unequivocally that problem solving is locally shaped. There is no argument about that. What we are interested in is whether (and how much) these current concepts and insights transfer to post local forms. In doing so, what we want to avoid is what Pinch (2008) eloquently describes as 'flip flop' forms of analysis. This is where scholars describe in one register the traditional face-to-face interaction and then 'flip' to another analytical frame to discuss the online form (see also Woolgar [2002]). On the one hand, we believe current approaches can be sufficiently extended to account for both worlds. However, on the other, it must also be emphasised that there are genuine differences in the way online technical support is organised that need to be accounted for. We start with the obvious point that what is at issue is no longer mechanical support but the working of an information infrastructure operating through extremely complex suites of software. In such instances, hypothetically and as it appears in practice, support staff no longer need to travel to physical sites to inspect and adjust technologies but can reproduce the systems of their clients on computer screens whilst sitting in their own offices. Clearly, the kinds of ICTs involved in delivering these forms of support provide new kinds of affordances. One challenge is for localisation approaches to address the interactions and exchanges possible in online forms of support. In particular these appear to relinquish (not entirely but in important ways nonetheless) the dependency of technical support upon the local situation.

3.2. Why Are Situations Local in the First Place?

Much of the sociological literature is (implicitly) informed by the notion of the 'situation' as a confined or restricted space. Goffman saw the situation as crucial in mediating social interaction. He defines the situation as "any physical area anywhere with which two or more persons find themselves in visual and aural range of another" (1981, p84: cited in Knorr Cetina & Bruegger 2002). Suchman defines situated actions as: "actions taken in the context of

particular, concrete circumstances" (1987, viii). With this she suggests that actions are always co-produced by humans interacting with the social and material conditions founds in their work settings (Henke 1999). The key word here is 'material'. Interestingly Goffman (1961) also saw the role of the material as important in mediating interactions. As Pinch (2008) notes, objects are centrally located within Goffman's work on 'role distance' (in particular with the ways in which children rode merry-go-rounds, first embracing the new role, and then letting go of the reins, looking around, and so on, so as to look cool, suggesting they were distancing themselves from this role). This suggests that technologies *are* important in situations or, rather, as we want to argue, that technologies enable *different* kinds of situations. In Orr's (1996) case, photocopiers, because they require mechanical repair, can be thought of as 'local objects'; they create highly localised episodes of repair (where co-location, proximity and so on, are required features). By contrast, in the case of software, and, in particular, the packaged enterprise systems discussed here, we want to show how these technologies are creating a very different space for repair, one where the situation is increasingly stretched out. Thus, and to be clear, we are not arguing that Orr's analysis is incorrect (quite the contrary, it appears that photocopier repair will for the foreseeable future remain a highly entangled affair). Rather, as the Geographer Ash Amin reminds us, we have to be awake to the possibility that technologies can constitute new kinds of settings (Amin & Cohendet 2005).

We find useful here discussions from within the Sociology of Scientific Knowledge (SSK) where localist conceptions of science have been subjected to critique (Shapin 1995; Harris 1998; Merz 1998; Turnbull 2000). Sceptical of the claim that knowledge diffuses because it is 'true', scholars sought to show how the universality of science was both an 'acquired quality' and a 'local affair'. They did this by emphasising the way facts were produced with reference to specific places and times, that they were the product of particular communities, and that there were tacit practices involved in their production (Knorr-Cetina 1981; Turnbull 2000). However the 'localist turn' within the study of science and technology has been criticised, most notably by Harris (1998, 295), who has argued that an overly confined notion of the 'local' or 'situation' is not useful: " [w]hether one speaks of 'local', 'situated', or 'embedded' knowledge, the implication is that the narrative is somehow confined to a small 'space' – if not in the literal sense of a geographical metric, then at least in the sense of restricted social, cultural, and temporal metrics". By contrast, whilst he advocates that scholars *should* focus on local situations, this should not be at the cost of studying what he terms 'extended practices':

> Situated knowledge may thus [be] studied in a rich social and cultural context without sacrificing…extended practices (i.e., the members as agents of a corporate network). Situating knowledge and its means of acquisition in the context of corporations allows knowledge production to be viewed both as 'local' and as 'distributed' without

privileging the former over the latter or, more generally, the micro over the macro (Harris 1998, 298-9).

We are in accord with these sentiments, and suggest that the conception of the situation of repair be extended to account for the fact it is no longer principally the physical situation (or place-based social relations) that bear the brunt of problem solving. It is for this reason we might think of the place in which repair takes place as an 'extended situation'. This has the benefits of retaining those aspects of Orr-type studies we find useful but also allowing scope for highlighting differences. This includes the vendor's choice to aggregate previously dispersed and territorial expertise into a global network of regional centres.

3.3. From Moving Solutions to Moving Problems

One other template we found for conceptualising this distribution of competence and subsequent search comes from the field of Organisation Science. In his research on 'telephone hotlines' Brian Pentland's studied two software vendors operating a 'telephone model' of support whereby help was almost entirely delivered remotely from specialised offices and by dedicated workers manning phones. Whilst sharing Orr's interest in situated work practice, Pentland also set in train an alternative view. When faced with complicated calls, for instance, support workers embarked on what he calls, drawing on Goffman, 'organizing moves'. That is, they did one of two things: they *moved the solution* through getting help; or they *moved the problem* through giving a call away to someone else in the same office. The former involved the 'quick question' aimed at co-located colleagues, inviting them to 'take a look'. For the latter, Pentland writes:

> [t]here were four generic kinds of moves through which this was accomplished: *assign*, *refer*, *transfer*, and *escalate*. These moves resembled the ways to get help, except that they involved assigning the responsibility for a call to someone else. The responsibility could shift back to the originator at a later time, but for the time being, the call would be someone else's problem (1992: 539, *our emphasis*).

Whilst building on this view of support, we propose to take it further through exploring the influence of technology in shaping the assignment and reassignment of problems (and the subsequent stretching out of the repair space).

3.3.1. Affordance and Mediation

While Pentland acknowledges materiality through analysing how technologies provide either 'constraints' or, adopting Gibson's (1977) term, 'affordances', objects like the telephone are seen as simply giving a more durable form to a capacity that *already* exists within the support

organisation.[5] By contrast, we argue that it is necessary to address the material dimensions of artefacts and properties. This includes attending to the uneven affordances that may be built-in to the particular structure and content of the complex software portal described below (Hutchby 2001). Importantly, this is not to resort to mechanistic materialism or technology determinism, and nor is it to deny the importance of discursive processes. In this latter respect, Woolgar and Grint (1991) have argued that we can never know the capacities of machines; only the beliefs circulated by various parties about these capacities.[6] Though there is value in their critique of naïve realist accounts which take at face value the claimed material properties of machines, Grint and Woolgar's ontologically relativist stance has been contested (Kling 1992; Williams and Edge 1996; McLoughlin 1999). For most, such a view remains unsatisfactory for the analysis of material artefacts and practices and the ways in which technologies define the spaces in which users are able to act (Akrich 1992; Latour 1992). In our view, and where we differentiate our paper from Pentland's, is that we think the ability to distribute problems is not a capacity solely residing within the vendor but it is one that has also been given a particular impetus and shape by the technology. To say this in different words, the technology allows the organisation to do things that were not previously feasible (Leonardi & Barley 2008). Moreover, whilst the notion of affordance is useful, and we will use it accordingly throughout, there is a similar but perhaps more developed notion that we will also use. This is the notion of 'mediation' developed within the Sociology of Technology. It captures both the materiality of technology but also how artefacts can direct actors away from original goals (Latour 1999). We adopt the term because it allows us to account not only for the intended but also unintended consequences of the technology.

4. Background and Research Methods

The vendor studied (which we identify throughout with the pseudonym 'SoftCo') is one of the largest and most successful producers of packaged information systems in the world. It has a very large, globally distributed user base. With many thousands of customers, it potentially receives several hundred requests for help each day. The vast majority of these will be deemed 'urgent' where the user is experiencing downtime and immediate support will have to be provided – often to far-flung places. Previously, when the vendor operated a territorial model of support, a user interacted with a specific regional office, which 'owned' a particular group of users (Orr 1996). Whilst a user may not have had particularly close relationships with individual

---

5 This is what is sometimes referred to as the 'sociological' view of technology: the artefact is merely a neutral tool that humans use in the way they see fit; it neither adds or detracts from our goals whilst using it (Latour 1999).

6 For Woolgar and Grint 'the nature and capacity of the technology remain essentially indeterminate' (1991, 370). Grint and Woolgar thus reject, on the grounds of the multiplicity of possible accounts about these technical properties, any attempt to engage with the material properties of artifacts (Woolgar and Grint 1991; Grint and Woolgar 1997).

support specialists, they would know to which office they 'belonged' (ibid.). Likewise, in turn, the dedicated support offices knew which users belonged to them. However, the types of products discussed here have a number of features making territorial forms of support difficult and expensive to sustain. It is resource intensive and, given the complexity of various systems, it would be hard to maintain the full range of requisite expertise in multiple regional offices (Jansen et al. 2006). It follows that particularly complex problems could mean hours on the phone chasing specialists in offices around the globe. For more than a decade, then, the vendor has attempted to find alternative ways to deliver support - the most recent of these being the introduction of the 'support portal'.

4.1. Globalised Face-to-Portal Forms of Support

Today, in order to enact a more global form of support, the vendor's labs are equipped with various technologies and tools allowing the vendor to integrate and co-ordinate its responses to users. The support portal has become crucial for everyone involved: it is the medium through which users present technical problems; it is the platform on which the vendor coordinates its internal response; and it is the mechanism by which SoftCo delivers help back to users. Through the introduction of the portal, the support process has been changed in two important ways. Firstly, the territorial links between the users and support specialists within regional support office are now 'attenuated' and replaced with a system that distributes problems to where there is expertise. The vendor has developed an organisational and technical solution in the shape of a complex online system allowing customers to access a network of global laboratories operating around the clock. Requests for help are submitted through an electronic portal to be picked up by one of a number of 'labs' located around the globe. The vendor has more than 2000 employees working on technical support alone. These are widely distributed throughout the world being located in over 40 different countries. Moreover, and importantly, if the specialists in these labs cannot resolve the problem – either because they lack specific expertise or are simply going home for the day – the problems would then be 'passed' to another vendor lab in search of expertise and support staff awake and at work. Secondly, support has shifted from face-to-face to principally 'face-to-portal' interactions (Knorr Cetina & Bruegger 2001). This means that the vendor no longer deals with physical but virtual users. In theory, problem solving no longer takes place in an entirely 'open' world but is increasingly pre-figured (i.e., the portal defines the various moves that can be made). This means that breakdowns are potentially no longer completely 'ill structured' but are framed, and the user is no longer completely 'in the wild' but is contained to some extent. To reflect how interactions are increasingly standardised we refer throughout to how the vendor has constructed the 'user in a portal' (Hughes *et al.* 1999).

4.2. The Endowed User

Whilst *globalised face-to-portal* forms of support and problem solving affords the vendor with new capabilities (such as potentially managing large volumes of requests for help), it also has a number of unwanted consequences. One issue is that interactions within the portal are 'canalised'. Not only does the vendor view the user through technology but also, in turn, the user interacts with the 'vendor in the portal'. At first sight, it is difficult to see how the technology allows the kinds of intense interactions, knowledge flows and careful management that Orr (1996) saw as essential for the surfacing and repair of user problems. However, to balance this tension, the vendor has responded in a quite unusual way. They have attempted to enact an 'endowed user'. Various modifications were introduced into the portal, which included a more sophisticated system dedicated to responding to user problems. The key innovation is that the portal is designed in such a way that it is the user who influences the nature and pace of the service (through setting so called 'priority states'). Setting a problem priority at 'very high', for instance, means a request attracts an 'immediate and continuous' response (as promised in legally enforceable Service Level Agreements). However, these endowments do not come without responsibilities. If a user is to wield influence then she must allow herself to be prompted and guided by these various processes of configuration (technologies and policies).

To summarise, we find that users are configured within the portal. Their relationship with support specialists is attenuated but they are also endowed with certain *prostheses.* As well as directing and channelling the nature of support, the vendor has handed over, and this is given material form by the technology, various rights and abilities. What is interesting to see is how the support staff reacts to this endowed user. What challenges do users, as owners of failing systems, but endowed with equipment, present in the technical support process?

4.3. Research Setting

We had obtained exceptionally good access to SoftCo which meant we were able to observe the support process from a number of distinctive viewpoints. Firstly, this was from 'inside' where one of the authors carried out an ethnographic study within a vendor lab for a period of five months. During this time, she worked as a support employee in SoftCo's 'Development Support' section, which is the highest level of technical support a user might meet. Her previous training in programming meant she was able to participate in the technical work. Thus, the bulk of her day was spent responding to messages and resolving problems reported in the portal. Based on the period of immersive study we are thus able to describe the general policies and practices found within this particular part of the organisation. Secondly, we also viewed the support process from 'outside' where we witnessed attempts by one user organisation to gain

help and expertise. Most of this material was gathered whilst one of the article's other authors was conducting research on a large institution based in the UK (which we identify throughout with the pseudonym 'UserOrg') implementing the vendor's Customer Relationship Management (CRM) system. Unfortunately for them (but fortuitously for us!) the UserOrg CRM system crashed during the period of research meaning we could also observe its attempts to gain help through the portal.

4.3.1. The North American Lab

Our fieldworker was based in one of the vendor's several 'Development Support' labs. The lab is spread over two floors of a multi-occupancy building located in the industrial area of a large cosmopolitan city in North America. This particular centre was responsible for developing and maintaining the vendor's CRM system (as well as a number of other technologies). Working in the 'support team', the fieldworker had her desk on the margins of the support area and close to the 'developers' allowing her to move between both communities.

Like many modern work environments, the majority of the employees spent the bulk of their day staring at computer screens. One of the things they were (obsessively) watching for was the arrival of 'messages' from around the world. These were the requests for help sent in by users through the portal. The support programmer close to where our fieldworker sat had a number of applications open on her computers through which she could view messages. For instance, a 'global monitor' presented a continually updated list of existing and new messages arriving in the vendor organisation. Messages arriving in the portal were glanced at, but no more. Support staff in Development Support knew these would be dealt with by colleagues 'lower down' the Product Support Chain. They mentally noted 'very high' priority messages for the modules for which they were responsible (since it was possible they could be passed to Development Support by the other support tiers within the next few hours) but took no action on them.

5. The Case Study

5.1. The Product Support Chain

We return to the Development Support lab in a moment but first we comment on the specific workings of what was called the 'Product Support Chain' since it provides a quite distinctive organisational form influencing the nature of vendor's response. Once reported in the portal a problem is received by the first link in the chain, by one of the 'Primary Support Centres'. There are approximately 20 of these providing a basic level of support. One specialist describes what happens to a message when it is received in these centres:

> …the message is immediately taken depending on the priority taken by someone wherever they are in the world, whoever is awake in the Primary Support Centres. A big

one is in India, Dublin and Spain. We have one in Asia Pacific, I think it's in Japan, one in China, in Australia, in America... Anyway, they take the messages. They have some kind of expertise depending on the component the message is assigned to. It might be the case that the customer has not read the documentation properly. And they just give them assistance on that. And then the message is solved. If Primary Support can't handle the message, then they send it to the next level (Development Support Specialist, email to fieldworker).

There are several interesting points in this quote, but we consider only the last one for the moment. If a programmer in the first part of the chain cannot resolve the problem it is passed 'vertically' up the chain to the next level, the Secondary Support Centres. Whilst staff in the first level were 'generalists', those in the next level had more specific expertise. During fieldwork, for instance, we saw how the vertical passing of messages from the first to second link was a highly effective way to weed out problems (and most were sifted in these initial stages). However if a message could not be resolved at these two stages then it was passed up to a third level called 'Development Support'. This is the final link in the Product Support Chain (and where our field worker was located) and there are six such 'labs' dotted around the world. Below, we present the passage of one specific message as it moves between various labs to indicate the nature of the journey it undertook. We demonstrate how a curious process had developed *where problems were sent off in search of expertise*.

5.2. Searching for Expertise

When UserOrg's newly implemented Customer Relationship Management solution crashed, it immediately submitted a 'very high' priority request for help within the portal. Their message read:

> We have recently gone live with a portal solution at [UserOrg]…The project deadlines did not allow sufficient time for testing all operating system/internet browser combinations that we need to support given that our portal is externally facing and can be accessed by any internet user. Since go live we have identified two serious problems with the Firefox browser which prohibits the submission of an online application to our portal by prospective applicants [-] hence we need a resolution as soon as possible.

As it was of the highest classification, the message was taken straight away by a Primary Support programmer. However, upon investigation, the problem appears to be complex so she immediately re-assigns the messages to a Secondary Support lab in another country. Having picked up the message a Secondary Support employee writes the following reply within the portal:

> Please also be aware that I'm not an expert in the issue & have limited knowledge in this area…in saying this I will try to resolve the issue… However, we may need to eventually pass this msg to Development Support, they are currently out of the office until 6am CET tomorrow. Please be aware that you may have to wait until then for expert help… (Jeff, 17:27 CET, 30th Nov).

This programmer points out how he lacks expertise in the particular system mentioned but says that he will 'try' to resolve it. He also sets up UserOrg for the possibility they might be passed to another lab and that they may have to 'wait' for help. This is indeed what happens. The message is picked up by another Secondary Support lab the same day with a programmer there attempting to fix the fault. However, he also has to admit that neither he nor his colleagues are experts either:

> As per telephone conversation, there are no CRM-IC experts available tonight. If none of the recommended notes help resolve the problem, please return the message to [SoftCo] for further analysis (Ethna, 19:13 CET, 30th Nov).

He makes a suggestion of a possible fix and advises the UserOrg team if these fail, to leave the status of the problem in the portal at 'very high' so that their message will be picked up shortly by another lab when those staff arrive in work. In fact, this is what happens, and the message is taken up by someone called 'Thomas' at a Development Support lab. The UserOrg team tell him they think their problem is linked to a particular software module (the 'ICWebClient'), but since Thomas is also not an expert on this module, and because it is a public holiday in his country, he passes the message on to specialists in another country (the team which specifically works on the ICWebClient):

> I forwarded your message to our development of…ICWebClient. One colleague from there will get in touch with you asap. Thanks for your understanding for this procedure (Thomas, 15:40 CET 1st Dec).

Before the message arrives however, it is picked up later that night by the 'Priority 1 Coordinator', who has responsibility for checking and following the journey of very high priority messages. He does not influence the direction of the message, checking only on the status and urgency of the problem:

> Called Graham…They are unable to save info on the CRM system which went live yesterday the 30th November. Customer needs to get this solved asap although they are willing to wait for the development team tomorrow morning (Sean, Priority 1 Coordinator, 19:26 CET 1st Dec).

In this case, since the user appears 'willing to wait for the development team', no action is taken. In the morning, and as promised, an ICWebClient team member contacts UserOrg. He asks them to amend the status of the message in the portal down from 'very high' to 'high' so that the problem will not be passed on any further. Indeed the message is held by this specialist for almost two weeks whilst he figures out a solution. However, in the meantime, there is another more serious failure in a different part of the UserOrg system and the team is required to submit another very high priority message within the portal. The whole process starts all over again. It is the beginning of a long series of interactions that finishes when UserOrg, frustrated

14

by the length of time it is taking to arrive at a fix, remove themselves from the portal through 'escalating'. We return to describe this escalation in a moment.

First, however, we note how these exchanges were organised around particular themes. In contrast to Orr-type studies where specialists travel out to clients, problems in the portal were sent off in search of expertise. Moreover, because of the initial openness of problems, it was often unclear as to where expertise might be found (this is said to be typical also of face-to-face modes of support – see McDonald & Ackerman [1998]). Indeed, many interchanges within the portal concerned the contingency of specialist expertise. Programmers were always careful to state at the outset whether or not they possessed sufficient knowledge to attempt a fix. Indeed, as is clear from above, the bulk of support staff only had a very general level of understanding of particular software components and thus communications with users could be often be quite humble ('I'm not an expert in this issue'), with programmers seemingly willing to advance a hypothesis (suggesting past 'fixes' and 'notes' within the portal to look at), but also not claiming definitive expertise (replies often highlighted the possibility that their suggestions may not work). There was mention of the possibility of further distribution; replies pointed to how messages could be assigned to another group of experts at any time in the near future. Many messages indicated temporal issues surrounding the search for expertise. Certain people, the 'Priority 1 Coordinator' for instance, were concerned only with managing the 'timeliness' of messages. Temporal issues surrounding problems were particularly acute because the user endowment introduced a dilemma. Support staff either sought out the correct competence but at the risk of not maintaining timeliness (asking users to 'wait' for help) or they maintained timeliness and distributed messages regardless of whether those receiving them possessed expertise. This latter issue often led to a practice known as 'ping pong' (see below).

To summarise, technical support has been recast and inserted into a new geographical and temporal regime where problems are distributed in search of expertise, or where messages are simply moved on because time dictates. However, whilst the vendor's relationship with the user is potentially attenuated, we see the introduction of various strategies to counteract this. It was a requirement (part of the endowment) that programmers regularly updated users about what was going on with their problem. This had the interesting effect of leaving support staff no choice but to share with users the various contingencies and uncertainties surrounding the online process. Thus, the portal turns out to be a relatively rich space where specialist staff offer up all kinds of information about the journey of messages – a form of knowledge we describe as 'portal knowledge'. Through this notion we refer to how all interactions are organised around the technology. Everything about a problem and its passage around the vendor sites was captured in the portal. This included information collected automatically by the system (i.e.,

how the portal clock measured the length of time a message remained with particular support staff) but also the fact support staff would ensure that all exchanges, suspicions and often 'feelings' were logged in the portal (see below). This often could include the laborious work of typing into the system any discussions held over the telephone.

5.3. Removing the 'Where' of Repair

One of the most notable outcomes of this globalised face-to-portal form of support is that whilst vendors need to attend to problems *when* they happen they are no longer seemingly concerned 'where' they occur. This is because they (potentially) have the means to disentangle problems from user contexts. When a message eventually found expertise, or simply came to the final link of the chain, a support expert would set about investigating and resolving the problem. What does investigating the problem mean in the context of the lab? A support specialist would read descriptions of the failure posted by the user, look at 'internal memos' in the portal describing fixes already attempted by other labs, and then set about resolving it herself. Typically, since these systems demanded tailoring and integration with existing information infrastructures, they would crash in the course of implementation due to some kind of unpredicted interaction with software components found at the user site. This meant it was extremely difficult to establish the provenance of a bug or diagnose reasons for failure. Was it a problem with the standard solution or has it occurred because of some changes attempted by the user? If a problem was found in SoftCo's standard technology then things were relatively straightforward (work would begin on a fix within the labs). However, if the bug was located in an area considered to be outside the standard software package then the problem was seen to be the users' responsibility. Diagnosis was not only the issue of pinpointing a fault but of setting its boundaries (i.e. who had responsibility for it). We might suggest these problems were 'sticky' (von Hippel 1988) because they initially resist separation from local context (Pollock 2005). The issue for the vendor was to transform problems from ones that were sticky to ones that could be accepted or rejected. The vendor took this kind of work seriously for it had no wish to be swamped by the many implementation problems experienced by each user. However, transforming a problem required effort, involving not only sophisticated strategies but also disciplined programmers to implement them.

5.3.1. Triage Devices

We show one example of how a message was rejected and a further one where, because a complication develops, other strategies had to be deployed. The quote below is typical of how problems might be rejected and we note how the programmer is careful not to decline the request outright but to offer truncated levels of support:

I gave you a few hints above. However, I would like to point out that I can only give support for the standard version. [User name] is responsible for its own customized version. Therefore, if the standard version is working, then I cannot offer support for [user name]'s version because it shows that the problem is not coming from our side (message from Development Support to user).

Even though some assistance is offered, it is made clear it is not sanctioned but informal help. Unsurprisingly, perhaps, refusing requests for help was often a highly complicated and sometimes contested issue. For instance, there were problems that resisted easy categorisation and remained between states - not clearly identified as belonging to either party – and thus might be described as 'liminal problems'. Liminal problems were potentially damaging for the vendor because even though there was uncertainty about the cause of a failure (whose responsibility it was) protocols dictated that it should be the vendor assuming liability (until proven otherwise). In practice, this meant that 'the clock' in the portal ticked against the vendor, and if the issue was not quickly resolved, a user could 'escalate' the matter further. The Service Level Agreement was key here. Under the vendor's recently launched 'maximum attention service', for instance, there was a guaranteed response time for the highest priority messages of not more than 'one hour'. After this, the vendor is committed to issue 'a correction, work-around or an action plan within the next four hours'. If the vendor breaks this promise, in theory, the user can ask for compensation (SoftCo document). It was rumoured in the lab that one user organisation had recently been awarded millions of dollars in lost revenues when SoftCo failed to deal with its problems in a timely fashion. Thus the vendor looked for ways to avoid being drawn into these kinds of difficulties and had put in place various 'triage devices' that could potentially sift difficult and time consuming problems. In the case of liminal problems, for instance, it had become standard practice to ask the user to run 'two systems' when conducting fault finding. This involved running data through their localised package *and* the standard solution whilst performing the series of steps causing the initial error. The 'two systems' strategy was a highly effective triage device because the user could seemingly see with her own eyes where the responsibility for a bug lay. If the problem occurred in her local system only, then the problem was clearly the user's; whilst if it could be reproduced in both, it was SoftCo's.

Yet interestingly, and despite this strategy, there were still difficult cases. One team of programmers, to which our fieldworker was attached, was involved in a lengthy exchange with one increasingly 'irate' user. As was normal practice, they had asked the user to supply information about how his problem occurred and to confirm he was running two systems. His replies were strangely ambiguous, however, and despite further questioning, no clear answer was forthcoming. As a result, support staff had to assume the problem was occurring in *their*

standard system. They continued with a large-scale examination of the generic package, but were still unable to reproduce the error. On further questioning, the user claimed the problem was still there but that it was 'intermittent' thus difficult to reproduce. The support specialists doubled their effort but to no avail. This went on for several days and, as time went by, the user was becoming more frustrated. He began to insist the problem be given more attention and to affect this he raised its priority in the portal to 'very high'. The person working on the bug could see an internal memo had been added within the portal:

> Dear Colleague, Customer called in to expedite the process of the message and the priority will raise to Very High upon customer request. The Production is not down but the system is greatly affected.

The team overseeing the problem became 'slightly nervous' as they were still unable to replicate the error and began to suspect an 'escalation' was forthcoming (which they feared, as we see below). Other experts were drafted in but they too were unable to resolve the issue. In a last ditch effort the programmers went back to the user asking to confirm he was indeed running both systems as requested and to state in which the error occurred. At this point, they receive a short reply where the user acknowledges the errors were occurring in his *local version* only! Responsibility thus immediately passes from SoftCo back to the user. However the days of wasted effort has caused some bitterness in the labs (since it was not the first time this user had attempted such a move). It was widely suspected that certain users would avoid directly responding to questions in the knowledge the vendor would give them the 'benefit of the doubt'. It seemed face-to-portal relationships created ambiguity and, on top of this, the fact that users were heavily endowed encouraged certain people to act opportunistically. Yet despite everything, the team leader advocates a 'friendly response' stating that 'even if we don't like them they are still our customers' (fieldwork diary).

To summarise, we have described how the reconstitution of technical support involves not only technology and organisation but also strategies to *transform user problems* at a distance. Orr-type suggestions that failures are inherently 'rooted' are clearly inadequate here. We have seen how bugs have been made to go from being 'unruly problems' in the local site to ones which can be distributed as 'problems in the portal'. Problems in the portal are 'light' meaning they are no longer tied down to particular localities. Disentangling problems required disciplined staff implementing various 'triage tests' as well as the ability to turn the distant and undisciplined user organisation into a 'laboratory' (Latour 1987) full of responsible actors. The user is handed part of the apparatus for demarcation: these endowments give to, or, to be more specific, attempt to produce in users certain responsibilities towards the self-management of problems. However, some users refuse to play the role allotted them and attempt to exploit the geographical

separation (or endowment) in order to claim more resources. In the following section, we turn our focus to how the support staff manage to live with the endowed user.

5.4. Life Within the Portal

The arrival of messages created the rhythm within the labs. Despite the sophisticated technologies available, the only way to check for new problems was to open the message inbox (which support staff did compulsively!). The specialists were always attentive to messages assigned to them. Requests for help prompted staff into action, dictated what they should work on, shaped the organisation of their working day. This rhythm was relentless. Once received, programmers *had* to respond to messages. Very high priority messages literally *challenged* programmers to act on them. The weight of a message was such that it was inconceivable it might be ignored. Once a message arrived, it would be opened and the programmer would add their name as the designated person responsible for the problem. A message was now 'in process by the vendor' where the portal would automatically log the time messages spent with SoftCo (and the particular support worker). This meant the programmer would either have to attempt to resolve it or find some other way to handle it.

5.4.1. Managing the Endowed User

Priority levels were rarely set in concrete. Users within the portal would raise them when they perceived a problem was not receiving the kind of attention they thought it deserved. Just as common, however, support staff might attempt to negotiate levels down. We say 'negotiate' because, once assigned, it was normally only users who could adjust a priority level. Since SoftCo had handed this ability to the user, programmers were unable to alter levels without first seeking the user's explicit agreement. There were exceptions, such as when it could be demonstrated a user was flouting their responsibilities, which meant a programmer was then endowed to make a change. However, this endowment had the effect of drawing staff into extensive negotiations with users about the exact nature and urgency of problems. For instance, when UserOrg submitted their message using the highest priority they immediately received the reply:

> You have logged this msg in Priority 'Very High'…Please be aware of the Priority & its restrictions re Note 67739. You say in the msg that this is affecting the 'go live' of the project. Can you please tell me when this is…? Also, please provide a 24hr contact number & open all relevant connections re note 67739….

Here the classification is queried since the formal definition states how there must be an 'absolute loss of the system' (which is not the case at UserOrg). The programmer also makes the user aware of their particular responsibilities regarding highest priority messages and directs them towards the relevant documentation in the portal. As the message describes they must now

open relevant connections (i.e. make the system ready so that the problem can be delocalized) and ensure that UserOrg staff are on site around the clock. When replying to the posting the UserOrg team defend their priority selection, stating:

> This is creating a huge problem in our production environment and is affecting a lot of our customers. All relevant connections are open… An early response would be highly appreciated.

The support specialist accepts the explanation and is therefore now unable to modify the status of the message:

> I [am] aware that you have gone live with CRM this morning and that this is an urgent issue. If the message is returned, I can leave the status at 'very high' and assure you that one of my colleagues will review it first thing tomorrow morning. [I]s this…OK with you[?].

The fact the message is accepted means a number of processes are set in train. However, in this instance, it is negotiated as to whether the response can wait until 'first thing tomorrow morning'. The user agrees without much ado. It seems in the face of these formal requirements, the working and to some extent private lives of the support staff depended on these micro negotiations. To demonstrate this we highlight a case where a specialist asks the UserOrg team to amend a message status not because he doubts the seriousness of their system failure but because it will mean he is able take a country specific public holiday:

> …today is holiday in [specific country] and I would ask you if [there are] these problems still…can [they] be postponed [until] tomorrow [and] to set the priority onto 'High'?

According to protocols, the problem can be 'postponed' for a day but only if the UserOrg team 'agree' and they must physically lower the level. There is the interesting issue of 'control' and 'materiality' here: it is the user herself who must authorise the change and then must enter the portal to modify the actual level otherwise it will continue to automatically record the amount of time a message is with Softco. A very high priority problem left overnight will attract attention from the Priority 1 Co-ordinator and lab managers (with possible sanctions being applied to the responsible vendor programmers). Interestingly, UserOrg does not respond immediately to the request. Thus a few days later a vendor programmer contacts them to ask if they can lower the message priority since the system, as far as he can see, now appears to be up and running:

> Because productive use now seems to be on http, we would like to reduce the issue's priority…

Again, as far as we understand, UserOrg does not respond. One specialist then telephones the UserOrg team and, because she does not find anyone there (whilst it is daytime for this programmer it is well into the evening for UserOrg), she writes in the portal:

> Graham is gone for the day. Not reachable…I will lower the priority. As described in the note 67739 a contact person needs to be available 24h.

One of the responsibilities that come with sending very high priority messages is that users must ensure that they have staff available to answer SoftCo questions whenever they might come. Since support is delivered from any one of the labs it could be that questions arise during the night. She points out how UserOrg were no longer maintaining their responsibility towards very high priority messages or conforming to the conventions of the portal; and *she* is then endowed according to vendor protocols to lower the priority level (which she does).

To summarise, the user appears to be at the centre of this new kind of distributed support. Enabled by the portal, she dictates the pace of the help received and, in turn, support staff find they have to negotiate the adequacy and immediacy of proposed solutions. However, support specialists also work hard to ensure the various conventions and norms of the portal are upheld. It is to this issue we now turn.

5.4.2. Portal Knowledge and Forms of Non-Passing

We have shown how within the portal greater attention is given to messages with a higher priority. This begs the question as to what happens when there are several messages with the same priority. When such 'stalemates' occurred, specialists applied an alternative logic. One programmer discussed how she would deal first with those users deemed to be the 'most important' to the vendor. Importance in this sense was related to whether they were operationally significant for the release of new products and modules (if the user was a 'reference site' or 'ramp-up customer' for instance) or more straightforward issue of hierarchy (whether the user was a prestigious firm or from a strategically significant sector) (Pollock & Williams 2009). One support programmer writes in an email:

> Of course if we get requirements from big [customer group], they have to be taken more serious[ly] then from very small ones. For ex[ample], if 5 of the big say they want something, that's going to be taken more seriously than if it's 2 small ones. But that's a law of the market. The bigger you are, the more attention you get (Support Programmer, email to fieldworker).

Other factors influencing speed of response were technical complexity or timing of the problem. For instance, none of the programmers liked to receive a difficult problem just before a weekend. In addition, more individual and personal factors seemed to play a role. There was the 'mood' of the programmer that day, how they 'felt' towards a particular user, or whether they actually 'liked' them. Interestingly, in this latter respect, personal or 'portal' knowledge of users, and their previous message practices, appeared to have a strong influence on how and when messages were dealt with. These included issues such as the 'politeness of the request', whether they had previously said 'thank you' for help received, or had left positive feedback.

Added to this, some users were known to 'exaggerate' and 'over classify' messages, or, as we have seen above, to seek help for which they were not entitled. These users were usually already known and they thus received a kind of implicit de-prioritisation of their problems. For instance, on receiving a high status message just prior to the weekend, one programmer said to another:

> Ah, that's [user name]. They always put their messages on high importance on a Friday afternoon even though it's not. We can leave that until Monday (fieldwork diary).

In a similar manner, a user who routinely over-prioritises her messages is ignored in the hope that she will learn to modify her behaviour:

> She always comes last minute. Maybe we should teach her not to do so by not helping her this time (fieldwork diary).

Some messages were also 'slowed' in the knowledge that the user in the past had been sluggish in providing all the necessary information (i.e. they had not maintained responsibilities towards very high priority messages):

> The Indians will take forever to respond anyway (fieldwork diary).

A message could be slowed in a number of ways. One common practice was to ask users for clarifications and further information about problems even when the answer was already suspected or known. Slowing worked as a strategy because while a clarification was expected from the user the message was officially designated in the portal as 'in progress by the user' and was thus not counted as time spent with SoftCo.

5.4.3. Unwanted Forms of Passing

In the article we have focused on the passing of technical problems (their 'exporting'). Allied to this, a more informal set of practices was common within the labs, which everyone described simply as 'ping pong'. Faced with the relentless arrival of messages, many of which might be complex and difficult to resolve, or were from users who were known to be 'trouble', programmers would sometimes ping-pong messages to other parts of the organisation, knowing full well that expertise might not be found there. It was simply sufficient that *their* problem would become (to paraphrase Pentland [1992]) someone *else's* problem. Indeed the most problematic failures, or troublesome of users, could be ping-ponged back and forth and across continents for several weeks. This was possible because the territorial links and issues of 'ownership', expressed so eloquently by Orr (1986), were clearly not present within the portal. Below, the 'Priority 1 Coordinator', who has responsibility for managing the timeliness of messages, describes how she picked up a message that had been travelling between labs for some time:

> The customer message was initially forwarded to the wrong component and subsequently went back and forth between the customer and [SoftCo] support with no

ownership taken for the message. Having received no information for a month, the customer requested to close the message as 'it was taking so long that I might as well just live with the bug as [SoftCo] is obviously not concerned in fixing their own bugs'. I picked up the message while monitoring messages for the specific market and contacted the customer directly. I listened to the customer's frustrations, understood the situation and promised to take action. While the customer appreciated the efforts he advised – 'you are flogging a dead horse' (Priority 1 Co-ordinator, email to fieldworker).

Whilst in this case the actions of the Priority 1 Coordinator meant the ping-ponged message finally found the appropriate expertise, the vendor was continuously attempting to reconfigure the portal through putting in place new roles, policies and practices to improve the message management process. We identify two projects implemented in this respect (and we analyse them as the emergence of a 'passing regime').

5.5. The Emergence of a 'Passing Regime'

5.5.1. New Technologies

When a programmer's inbox was full of same priority messages there could be some uncertainty about which user should receive help first. As we mentioned above, programmers would typically pick through messages applying their own discretion. However, SoftCo continuously sought to nudge the problem sorting process from the realm of individual discretion and towards a more formal and technologically managed decision-making process. A new piece of software was therefore introduced within the portal to automatically prioritise and rank stalemates according to a number of pre-defined criteria. One support programmer describes this system in an email to the fieldworker:

> …for prioritizing messages, we're now using a new monitor that prioritizes the messages for us according to a number of factors (agreement with customers, time spent at [SoftCo], priority). This way, when you have a ton of messages, it's now much more easier to select which one to work on… (Support Programmer, email to fieldworker).

We see here a form of technical configuration or mediation where a further technology is inserted between the programmers and users, which the support staff were encouraged to take into account. The new monitor will take on the role previously enjoyed by the (apparently undisciplined) programmers of scheduling support work. The vendor wanted users to be dealt with in the order it thought fit and thus used a technology to compensate for what might be described as the 'strategic failings' of its staff. When asked if she thought the new technology took away her ability to schedule her work, the specialist from above was clear it did not:

> I don't feel it takes freedom away from us. The messages are prioritized but we still pick the ones we want to work on. It's just that [SoftCo's] prioritization method is clearly shown...I just use the tool. I just don't question their rules (Support Programmer, email to fieldworker).

For her, at least, the tool was not a concern as she thought it would offer her greater clarity when choosing between problems. It equipped her with the ability to make decisions that were broadly in line with her employer's strategy and preferences and she saw this in a positive light.

5.5.2. Bureaucratic Forms of Ping-pong

The second innovation concerned the practice known as ping-pong which, whilst long established in the labs, was now increasingly problematised. An internal communication was released stating ping-pong was no longer to be tolerated since it 'significantly damaged the vendor-user relationship' (SoftCo, internal document). However, ping-pong was potentially difficult to control since it closely resembled the vendor's own strategy and means of dealing with the user problems: passing and ping-pong were two sides of the same coin (a new order produces its own disorder [Berg and Timmermans 2000]). The vendor response was thus not to outlaw ping-pong (this would call into question the organising principle of the support process) but to more closely regulate it. Managers raised a protocol that stipulated just how and when users could be passed. Above all, it emphasised the need for a 'justified reason' to be given. Rather than simply re-assign a message to a lab or group elsewhere, the new protocol required a programmer to demonstrate how a user problem connected to modules other than the ones for which they had responsibility. Such reasons might include where a message was received which clearly concerned the module of another team or lab. In this case, it would be considered legitimate to pass the message onwards. However, before this could happen, everyone involved in what was described as the 'passing chain', would have to agree on the necessity of redistribution.

Unsurprisingly, perhaps, and despite attempts to control ping-pong, it was still commonplace, though it did take on a more complicated, bureaucratic form ('regulated ping-pong').

5.6. Overflowing the Portal

A further consequence of the practice of passing was that this form of virtualised support was prone to break down in quite dramatic fashion. Users would sometimes find themselves having to 'escalate' their problems out of the narrow, restricted and virtualised world of the portal and back into the vendor organisation.

5.6.1. Escalation is the Word

The setting of message priority levels was not the only way users were endowed. If a user found she was being ping-ponged around the organisation, or was simply unhappy with the length of time her problem was taking, she could 'escalate' her problem. Escalations were a major event within the vendor and were reportedly 'feared' by staff. In the five months spent at the vendor, our fieldworker never actually saw an escalation but (as described above) her team did once

have a 'near escalation' (which was a major event in itself!). Our fieldworker was repeatedly told that 'nobody wanted an escalation on their shoulders' since it meant 'the board' would from then on be closely monitoring them (and this could apparently be damaging for a programmer's promotion and career prospects). Escalations differ from the earlier form of endowment in that they are not selected in a drop down menu. Instead the word itself has to be verbally spoken or written. When UserOrg, for instance, found their problems were not being resolved – they had been in the portal for several weeks with little progress – they escalated:

> We are concerned that the development period is continuing to stretch and that this is compressing the already inadequate time we have for testing, UAT and training…We must meet this target of 16 September however there are a number of concerns: a) the CRM middleware 'show-stopper' issue remains unresolved. This is now back from [the portal] but if we need to put it back into [the portal] we would appreciate help in *escalating* this (email from UserOrg to SoftCo, *our emphasis*).

Once escalated the nature of the problem resolution process changed dramatically. Vendor documentation describes how, after an escalation, the user must receive the 'immediate and maximum attention of the vendor until the issue had been resolved'. One support manager describes how seriously they take an escalation:

> We don't work on anything else. That's what escalation means. It doesn't necessarily mean that we have to be there at three in the morning. I haven't done it in four-five years. But it could happen that they say, sorry you have to work until it's solved (author interview with support manager).

Upon receiving an escalation, the following organisational processes were enacted. The support specialist responsible would contact their line manager who then sets about establishing a 'judging team' to investigate the issue. The first task of the judging team, made up of senior programmers, is to accept or reject the escalation. Unlike other kinds of endowment, the ability to escalate is contingent; it is the vendor's judgement as to whether an escalation is appropriate or not. It was explained to the fieldworker that the vendor maintained this right to avoid users 'over emphasising' their problems and obtaining valuable resources for themselves. In this connection, SoftCo had a formal definition for when an escalation could be accepted:

> …a major situation that causes a substantial negative business impact for the customer or [SoftCo]. A customer project must be escalated if the customer's [SoftCo] based solution has or will have: system standstill; server performance issues; functional gaps; implementation issues; operational issues; public relation issues; legal or compensation issues (SoftCo, internal document).

If accepted the problem is either classified as a 'message' or 'customer' escalation. The former refers to a specific instance where the user's systems are critically under threat but the latter is considered more serious since the relationship with the user is seen itself to be in jeopardy. We focus on one such customer escalation below.

5.6.2. Customer Escalation

Frustrated with the lack of attention their system was receiving, and having already attempted to escalate the matter, UserOrg now decide to bypass the support organisation altogether and bring their problem to the attention of SoftCo's executive management. A senior manager from UserOrg writes directly to the President of SoftCo's Global Public Services demanding something more substantial is done to rescue their system implementation, where a number of interconnected problems had begun to mount up. When working with a customer escalation the vendor appoints an 'escalation manager' who 'pays special attention' to the user and acts as the intermediary between the various actors concerned. One vendor manager from the labs describes what happens during such an escalation:

> If it is a formal escalation, it is taken care of by an escalation manager. The escalation manager will be close [to] the location where the customer is. If the customer is in US east cost, we have an escalation manager for him. The escalation manager is then going to check why they are escalating…. What that means, we pay special attention to this specific customer (interview with Vendor manager).

Interestingly, whilst in normal problem-solving circumstances geographical location or personal relationships were not thought to be relevant, this is clearly not the case with escalations. Here, the vendor appoints a senior employee with knowledge of, or relationships with, as well as proximity to, the user. This actor turns out to be crucial as she designs a 'de-escalation plan', listing the 'top issues' to be resolved. The various programmers working on the project will keep her up to date on progress, and she will regularly relay this information back to the user. The episode demonstrates how, when escalations occur, the vendor is forced to deal not with the manageable problems and disciplined users it has spent much time enacting but with 'unruly problems' and 'users in the wild' once again. Thus, once a user has overflowed the portal into the organisation, the vendor has little alternative but to revert to territorial Orr-type form of support (face-to-face, personal relationships, *etc*.). Within the scope of a customer escalation, then, technical support is once more an issue of proximity and close interactions.

This is not to suggest, however, that the vendor simply accepts this seeming limitation. Instead, they make every effort to find ways to 'systematise' such overflows or to find ways to avoid them. Programmers in Development Support, for instance, pointed out how most escalations did not occur 'out of the blue'. Some said that with experience they could detect 'warning lights' where a user was giving out 'strong hints' that she was about to escalate. If identified, these signs were documented within the portal, so that programmers were aware of how 'critical' the problem was becoming. Some even went as far as putting a small candle like symbol next to particular users in the portal to signal there was a potential escalation (and that programmers

should 'take care' when communicating). We see this as another example of the kinds of 'portal knowledge' mentioned previously.

6. Discussion

We began this essay with an apparent contradiction. Micro-sociological and interactionist based theories of technology tell us that technical support is an intrinsically *local* and *rooted* activity. Nonetheless, as we have shown, vendors have seemingly succeeded in moving the technical repair of their products online. Moreover, this is not the repair of simple technologies but some of the most complicated systems used within organisations today. This latter development appears rather 'unlikely' or 'unfeasible' through the lens of prevalent critical interactionist approaches, which emerged from a concern to counter the discourses of ICT-enabled delocalisation (for example from Castells 2000; Cairncross 2001). However, despite ongoing and significant developments in new information & communication technologies, micro-sociological accounts have continued to give primacy to direct interpersonal interactions and have implicitly portrayed more virtualised forms of interchange as lacking the required kinds of contextual knowledge needed to effect technical repair. This critical reaction against modernist claims has imparted a dualistic structure to much contemporary academic discussions reflected in a series of polarised debates about the respective roles of the local/global (Walsham 2001b), informal/formal (Berg 1998), embedded/disembedded (Giddens 1990), micro/macro (Knorr Cetina 1981), virtual/real (Woolgar 2002), and so on. The time has come to go beyond this kind of dichotomised debate. In particular we attempt to redress the shortcomings of those approaches which seem to be underpinned by the presumed victory of the 'the local' over 'the global'. A particular problem with these is that they fail to properly account for the new kinds of interactional possibilities (as well as unintended consequences) created by the introduction of new information and communication technologies.

To be clear, we are not simply suggesting a diminishing role for place in discussions of online support; our argument is more nuanced than that. The 'where' of repair still matters (Henke & Gieryn 2007) though it is now highly mediated by technology. Nor are we recommending the wholesale jettisoning of current localist approaches. Rather we argue for their extension to account for what are genuinely different modes for organising technical repair. Our starting point was Orr's (1996) still relevant study of communitarian based expertise and situated forms of action. We also sought insights from Pentland's (1992) account, inspired by Goffman, of the 'moves' and 'strategies' performed by support staff in managing and distributing difficult problems. Some elements of the shifts in the organisation of repair work described here – particularly in terms of how problems are passed around in search of expertise – were already

27

noted by these scholars. Orr characterised how the knowledge networks within which technical problems were resolved were by necessity continually extending outwards. Technicians did not always solve problems by themselves but often through moving within communities where they gathered and disseminated 'war stories'. This was developed by Pentland who noted that it was not simply knowledge but also *solutions* that were being moved around (i.e., support staff routinely brought other specialists into the diagnosis process). He also suggested *problems* too could be passed on (staff would shift particularly difficult or troublesome problems onto other colleagues in the same office).

In this paper we have sought to exploit these important insights by bringing into the picture new technology, how it *affords* particular capabilities and *mediates* certain kinds of interactions. Yet the idea of an *unmediated* setting is itself something of a fiction. Orr (1996) himself saw the situation made up of a triangle of relationships which included the technicians, the customers and crucially the 'machines'. Technicians already interacted with settings through technology: "Technicians' interaction with the machines as technical objects colors, mandates, and sustains the interactions with customers, managers, and other technicians" (Orr, 1996, 4). What we are attempting to do here is to show how the interactions – now that they are stretched out over an extended situation - are 'coloured', 'mandated', and 'sustained' through the introduction of new kinds of information and communication technology. That is, the portal defines (or bounds) the situation in various ways.

In particular we have shown how the diagnosis of technical failures, the identification of problems with the operation of certain technologies and the search for solutions are no longer rooted in the place in which they occur but can be *uprooted* and *passed* elsewhere. In the case described here, problems could be passed around a vendor's various remote sites in search of expertise and specialists who are awake and in work. Importantly, the disentangling and exporting of problems does not happen easily but requires much effort. It required the development and application of various techniques and technologies (configured users, disciplined support staff, triage devices, and so on). Moreover, we have shown that, whilst these evolving virtualised forms of technical support appear to provide a practical solution for many kinds of technical failure, they also had a series of perverse consequences. These new kinds of technology-*mediated* technical support opened up a novel socio-technical terrain with affordances which changed the support process in new and often unexpected ways (see Table One for a summary of some of the differences in relation to traditional and earlier forms of repair).

One issue where our work begins to contrast with existing scholarship is that, once online, the strong territorial ties identified by Orr (1996) seemed no longer to be essential. Relationships are attenuated, meaning support staff are no longer responsible for particular users. This meant particularly difficult problems or difficult users could be 'ping ponged' to other parts of the organisation. It is when this happens that we can say that the situation is genuinely extended; problems are delocalised (and as we saw, support staff and organisational users both sought to use this to their advantage). It is also important to note that this extreme form of extension is facilitated by the technology and the wider organisational infrastructure and processes that have been put in place (the technology affords the possibility to the support worker to pass messages on, and, in turn, the support programmer accepts that possibility).

A related feature was the development of what we have termed 'portal knowledge'. This included knowledge (information, suspicions, feelings, and so on) typed into, or collected automatically by the portal. Whilst aspects of portal knowledge might appear similar to Orr's 'war stories' (1996), there are also differences. This is that this information is made available simultaneously and instantly to various labs around the globe (this is clearly a new kind of affordance that scholars need to take in to account). The notion of portal knowledge has another aspect (which can also be found in Pentland's discussion of telephone support). This was that the geographical separation between users and the support labs encouraged opportunistic behaviour (for example, some users exaggerated problems to win for resources for themselves). The various participants gradually learnt how the portal worked and the various 'moves' (a la Pentland/Goffman) that could be made. And as a result many participants turned out to be quite strategic about their actions. Users found they could attract resources and attention by increasing the priority attached to their problems. One counter to this was that support staff, in turn, developed their own strategies to regulate behaviour within the support process (including 'slowing' responses to teach users a lesson, 'ping-ponging' problems elsewhere, and so on).

Interestingly, these counter strategies sometimes threatened the seemingly 'contained' nature of the support portal. Problems could *overflow* out of the virtualised domain and back into the physical organisation (for example through the mechanism of an 'escalation'). Such overflows were treated seriously. They were seen to be potentially commercially damaging. They also represented a 'failure' of online support. In these cases, the vendor was forced to retreat back to more territorial and location-based forms of help. We further observed how the vendor sought to avoid such episodes through continuously regulating and modifying support (a form of work that we have analysed as the emergence of a 'passing regime'). We saw how interactions between support specialists and users were further mediated by the introduction of new technologies (a tool to rank users by 'importance'). In addition, the practice of 'ping-pong' was

problematised. What is particularly interesting about this case (which may be unusual) is that the principle means by which this passing of messages was coordinated was through the enactment of the 'endowed user' in which users were able to command attention but in the same process took on certain kinds of responsibility. It is perhaps in this respect that our description of technical support begins to differ significantly from previous work.

Life in the portal presented various tensions for users and support staff alike. In terms of the former, their relationship to the vendor was 'canalised' and instead of working with specific support employees or a fixed office, their problems could now be passed around the globe to be dealt with where and when the vendor saw fit. To counteract this, the user was given the ability to 'self manage' within the portal, through being endowed with various prostheses. Technological endowments (backed up by legally enforceable Service Level Agreements) had the effect of putting the user at the centre of the process. Thus the 'where' or, more accurately, the 'time' of the user still clearly played a role. It was built into the very heart of the same technology attempting to uproot problems. As a result of the particular shape of the portal, the support staff were under various pressures (most notably temporal ones in which the portal 'timed' their actions) to respond adequately to the demands of local sites. Thus, to beat the portal clock, support staff had to negotiate 'extra time' or request permission for 'slight delays' from users (again putting the local site at the centre of the process).

Thus it can be seen that the portal embodies an interesting contradiction: it encourages disentangling, such that the 'where' of repair does not seemingly enter into the process, but at the same time it returns a focus on particular settings of repair. Endowed users (pending the raising of priority levels or escalating) are still highly delocalised users but ones able to exercise more direct control over the resources available to them, including control over the support staff working on their case. Enacting endowment rights creates a space in which support staff find themselves in a concrete and accountable relationship with the user (though ping-pong by support staff shows there was still non-sanctioned forms of behaviour here). It seems the endowment creates a limited and temporary arena where there is a relatively arbitrary and emergent relationship between particular support staff and settings. This is a new plane in cyberspace where the technology mediates an interesting kind of role, and it is one which signals a significant difference between us and earlier studies. Many of the 'moves' described here are also found in their work of Orr (1996) and Pentland (1992), but the difference in our case is that it is now the 'endowed user' who performs more of them. So whilst the game and strategies are similar they are performed by actors playing roles in a different (extended) situation.

7. Conclusions

To conclude, we have seen changes to the *nature* and *practice* of technical support as well as the *time* and *situation* in which it occurs. In contrast to the widespread polarisation of accounts between those which emphasise either the technologically-enabled abolition of the local or the continued importance of face-to-face interaction, we have explored how the difficulties of disembedding and diagnosing faults are addressed in new ways. However, they are only partially and incompletely unpicked. The attempt to move from territorial to global virtual – produces a partial resolution but in turn generates tensions and contradictions – which in turn become subject to further technical and organisational reforms. We have sought in this article to contribute to the more effective conceptualisation of these post local developments. If the study of repair has been characterised in the past by notions such as *rooted* and *embedded* problems and *localised* situations it should now be augmented with those of *disentangling*, *exporting* and the *globalised* or *extended* situation. In this latter respect, what we have described here is another instance of the new landscapes of interaction being opened up by ICT systems (Crang et al. 1999). This is that technologies allow for new forms of interactions but also different kinds of situations (Amin & Cohendet 2005). Whilst the technologies described by Orr (1996) create highly 'local' situations, the ones depicted here enable quite different spaces for repair (see Table Two for a summary of the various ways in which the portal constitutes the technical support situation).

7.1. Future Directions

The material discussed here represents a substantial reconfiguration of the technical support process. This suggests the need for scholars to reconsider not only the sociology of technical support but also, perhaps, a *new sociology of the extended situation*. A possible research agenda might be to consider in what ways the nature and role of actors within extended situations differ from more traditional ones. Orr's (1996) conceptualisation of a relatively simple triangle of relationships (users, technicians and machines – with only a minor role given to other nonhuman actors like service manuals) all interacting within a single arena may underplay the very different texture of the fabric of social relations and range of actors when deployed to more complex cases. We, by contrast, describe an elaborate and tangled space where there is a wider and more heterogeneous and distributed set of actors, linked together by, and working on, information and communication technologies, all within an extended fabric of individuals, organisations and globally dispersed laboratories. Questions for future investigations might focus on how (and in what ways) this distribution/heterogeneity changes interactions and relationships (whether this be in technical support or any other kind of ICT mediated technical

work). For instance, do we see changes in the roles and moves available to actors; do we see changes in the range of actors? Particular insights may be possible from examining how these technologically-mediated extended situations are created and evolve as the actors involved learn about their affordances and seek to exploit as well as attempt to regulate and control their operation.

References

Akrich, M. (1992) 'The De-Scription of Technical Objects' in W.Bijker and J.Law (eds) *Shaping Technology/Building Society: Studies in Sociotechnical Change*, Cambridge, MA: MIT Press.

Amin, A. and Cohendet, P. (2005) 'Geographies of Knowledge Formation in Firms', *Industry and Innovation*, 12, 4: 456-86.

Aneesh, A. (2006) *Virtual Migration: The Programming of Globalization*, Durham and London: Duke University Press.

Barley, S. (1996) 'Technicians in the Workplace: Ethnographic Evidence for Bringing Work into Organization Studies', *Administrative Science Quarterly*, 41: 404-41.

Bechky, B. (2006) 'Talking About Machines, Thick Description, and Knowledge Work', *Organization Studies*, 27, 12: 1757-768.

Berg, M. (1997) Of Forms, Containers, and The Electronic Medical Record: Some Tools For a Sociology of The Formal, *Science, Technology, & Human Values,* 22, 4: 403–33.

Berg, M. and Timmermans, S (2000) Orders and Their Others: On the Constitution of Universalities in Medical Work, *Configurations,* 8, 1: 31-61.

Cairncross, F. (2001) *The Death of Distance: How the Communications Revolution Is Changing Our Lives*, Harvard Business School Press.

Castells, M. (2000). 'Grassrooting the Space of Flows' in Y.Aoyama, B.Warf and J.Wheeler (eds) *Cities in the Telecommunications Age: The Fracturing of Geographies*, New York: Routledge.

Cowan, R. and Foray, D. (1997) 'The Economics of Codification and the Diffusion of Knowledge', *Industrial and Corporate Change*, 9, 2: 211-53.

Crang, M., Crang, P., and May, J. (1999) *Virtual Geographies: Bodies, Space and Relations*, London: Routledge.

Gable, G. (2001) 'Large Packaged Application Software Maintenance: A Research Framework', *Journal of Software Maintenance and Evolution: Research and Practice*, 13: 351-71.

Gibson, J. (1977) 'The Theory of Affordances' in R.Shaw & J.Bransford (eds.), *Perceiving, Acting, and Knowing,* Lawrence Erlbaum Associates, Hillsdale, NJ.

Giddens, A. (1990) *The Consequences of Modernity*, Stanford: Stanford University Press.

Goffman, E. (1961) *Encounters: Two Studies in the Sociology of Interaction*, Indianapolis: Bobbs-Merrill.

Goffman E. (1981) *Forms of Talk*, Philadelphia: University of Pennsylvania Press.

Grint, K. and Woolgar, S. (1997) *The Machine at Work*, Cambridge: Polity.

Harris, S. (1998) 'Long-Distance Corporations, Big Sciences, and the Geography of Knowledge', *Configuration*s, 6, 2: 269-304.

Henke, C. (1999) 'The Mechanics of Workplace Order: Toward a Sociology of Repair', *Berkeley Journal of Sociology*, 44: 55-81.

Henke, C. and Gieryn, T. (2007) 'Sites of Scientific Practice: The Enduring Importance of Place', in E.Hackett, O.Amsterdamska, M.Lynch and J.Wajcman (eds), *New Handbook of Science and Technology Studies*, Boston, Mass: MIT Press, pp 353-376.

Hirt, S.G. and Swanson, E.B. (2001) 'Emergent Maintenance of ERP: New Roles and Relationships', *Journal of Software Maintenance and Evolution: Research and Practice*, 13: 373-98.

Hughes, J., O'Brien, J., Randall, D., Rodden, T., Rouncefield, M. and Tolmie, P. (1999) 'Getting to Know the 'Customer in the Machine''', Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work: 30–9.

Hutchby, I. (2001) 'Technologies, Texts and Affordances', *Sociology*, 35, 2: 441-56.

Jansen, S. Ballintijn, G. Brinkkemper, S. and van Nieuwland, A. (2006) 'Integrated Development and Maintenance for the Release, Delivery, Deployment, and Customization of Product Software: A Case Study in Mass-market ERP Software', *Journal of Software Maintenance and Evolution: Research and Practice*, 18, 2: 133-51.

Kling, R. (1992) 'Audiences, Narratives and Human Values in Social Studies of Technology', *Science, Technology and Human Values*, 17, 3: 349–65.

Knorr-Cetina, K.D. (1981) 'Introduction: The Micro-Sociological Challenge of Macro-Sociology: Towards a Reconstruction of Social Theory and Methodology', in K.D. Knorr-

Cetina and A.V. Cicourel (eds) Advances in Social Theory and Methodology: Toward an Integration of Micro- and Macro-Sociologies, Boston, MA: Routledge and Kegan Paul.

Knorr Cetina, K.D. and Bruegger, U. (2002) 'Global Microstructures: The Virtual Societies of Financial Markets', *American Journal of Sociology*, 107, 4: 905-50.

Latour, B. (1987) *Science in Action, How to Follow Scientists and Engineers Through Society*, Cambridge, MA: Harvard University Press.

Latour, B. (1992) 'Where are the Missing Masses? Sociology of a Few Mundane Artefacts' in W.Bijker and J. Law (eds) *Shaping Technology/Building Society: Studies in Sociotechnical Change*, Cambridge, MA: MIT Press.

Latour, B. (1999) *Pandora's Hope: Essays on the Reality of Science Studies*, Cambridge, MA: Harvard University Press.

Leonardi, P.M., and Barley, S.R. (2008) 'Materiality and Change: Challenges to Building Better Theory About Technology and Organizing', *Information and Organization*, 18, 3: 159-76.

McDonald, D. & Ackerman, M. (1998) 'Just Talk to Me: A Field Study of Expertise Location', Proceedings of the 1998 ACM Conference on Computer Supported Cooperative work, November 14-18, Seattle, Washington, US: 315-324.

Merz, M. (1998) ''Nobody Can Force You When You Are Across the Ocean': Face to Face and E-Mail Exchanges Between Theoretical Physicists' in C.Smith & J.Agar (eds.) *Making Space for Science: Territorial Themes in the Shaping of Knowledge*, London: MacMillan: 313-29.

Nicholson, B. and Sahay, S. (2004) 'Embedded Knowledge and Offshore Software Development', *Information and Organization*, 14: 329-65.

Nicolini, D. (2007) 'Stretching Out and Expanding Work Practice in Time and Space: The Case of Telemedicine', *Human Relations*, 60, 6: 889-920.

Ophir, A. and Shapin, S. (1991) 'The Place of Knowledge: A Methodological Survey', *Science in Context*, 4: 3–21.

Orlikowski, W. (1996) 'Evolving with Notes: Organizational Change Around Groupware Technology' in C.Ciborra (ed), *Groupware and Teamwork: Invisible Aid or Technical Hindrance?*, Wiley: Chichester.

Orr, J. (1996) Talking About Machines: An Ethnography of a Modern Job, Ithaca: Cornell University Press.

Pentland, B. (1991) 'Making the Right Moves: Towards a Social Grammar of Software Support Hot Lines', Unpublished PhD Manuscript, Sloan School of Management, Massachusetts Institute of Technology.

Pentland, B. (1992) 'Organizing Moves in Software Support Hot Lines', *Administrative Science Quarterly*, 37, 4: 527-48.

Pentland, B. T. (1995) 'Read Me What It Says On Your Screen: The Interpretative Problem in Technical Service Work', *Technology Studies*, 2, 1: 50-79.

Pentland, B. (1997) 'Bleeding Edge Epistemology: Practical Problem Solving in Software Support Hot Lines' in S.Barley and J.Orr (eds) *Between Craft and Science: Technical Work in U.S. Settings*, Ithaca, NY: ILR Press.

Pinch, T. (2008) 'Where is the Goffman of the Internet?', Paper Presented at the Institute for the Study of Science, Technology and Innovation (ISSTI), University of Edinburgh, Edinburgh, UK, June.

Pollock, N. (2005) 'When is a Work-around? Conflict and Struggle in Computer Systems Development', *Science, Technology and Human Values*, 30, 4: 496-514.

Pollock, N. and Williams, R. (2009) *Software and Organisations: The Biography of the Enterprise-wide System or How SAP Conquered the World*, Abingdon: Routledge

Shapin, S. (1995) 'Here and Everywhere: Sociology of Scientific Knowledge', *Annual Review of Sociology,* 21:289–321.

Suchman, L. (1987) *Plans and Situated Actions: The Problem of Human-Machine Communication*, Cambridge: Cambridge University Press.

Turnbull, D. (2000) *Masons, Tricksters and Cartographers*, London: Routledge.

von Hippel, E. (1988) *The Sources of Innovation*, Oxford: Oxford University Press.

Walsham, G. (2001a) Knowledge Management: The Benefits and Limitations of Computer Systems, *European Management Journal*, 19, 6: 599-608.

Walsham, G. (2001b) *Making a World of Difference: IT in a Global Context*, Chichester: Wiley.

Williams, R. and Edge, D. (1996) 'The Social Shaping of Technology', *Research Policy*, 25: 865–99.

Woolgar, S. (2002) *Virtual Society. Technology, Hyperbole, Reality*, Oxford University Press, Oxford.

Woolgar, S. and Grint, K. (1991) 'Computers and the Transformation of Social Analysis',
   Science, Technology, and Human Values, 16, 3: 368–78.

Table One: Similarities/differences in Technical Support

	Orr-type	Pentland	Pollock et al.
Form of Technological Mediation	Extensively face-to-face	Primarily telephone-based	Predominately face-to-portal
How Problems are Managed	Problems assigned by territory	Problems assigned by functional area	Problems passed to whoever is awake and in work Problems advance through different levels
Mode of Problem Solving	Site visits and 'situated understanding' ('being there')	Experience based	Problems sent off in 'search' of expertise
Mode of Problem Diagnosis	Talk to customer Interact with machine Run tests	Talk to customers (ask them to 'read out' error logs) Assess customer knowledge	Talk to customers via portal (or telephone) Interact remotely with user system and run tests
Interaction between Colleagues	Fundamentally communitarian ('being together' and swapping 'war stories')	Principally communitarian (asking co-located colleagues 'quick questions' or to 'take a look')	Weak-communitarian (exchanging and contributing to 'portal-knowledge')
Ownership of Problems	Territorial (technicians 'owned' territories; providing higher level of support to 'their' users)	Organisational division of labour (specialists own problems)	Problems not 'owned' but assigned; attachments to users formed through maintenance of portal norms
Control of Problems	Users in the wild (technicians have to 'react to whatever they find at sites)	Users interactions limited (channelled through telephone)	Users are framed ('user in the portal') but also 'endowed'
Transfer of Problems	Competence-based (technicians call in help from community of practitioners when lacking expertise)	Technicians 'refer', 'transfer' or escalate problems. Referrals done by telephone operator; transfer agreed through face-to-face interactions between technicians; escalations conducted through seeking 'approval' from supervisors	Problems are 'passed' or 'ping-ponged' within portal Passing influenced by programmer discretion and user endowment (i.e. ability to raise priority levels and escalate).

Table Two: How the Portal Shapes Technical Support

Technology	Mediating effect
Portal as an active participant	It is a 'buffer' or 'broker' (cf. Barley 1996) within the situation
Portal as a container (the 'user in the portal')	It holds or configures users but is liable to 'overflow'
Portal as an arena for games	Problems can be exaggerated or ping-ponged
Portal as a world	"Life in the portal…" (i.e., programmers work according to the demands of the portal clock)