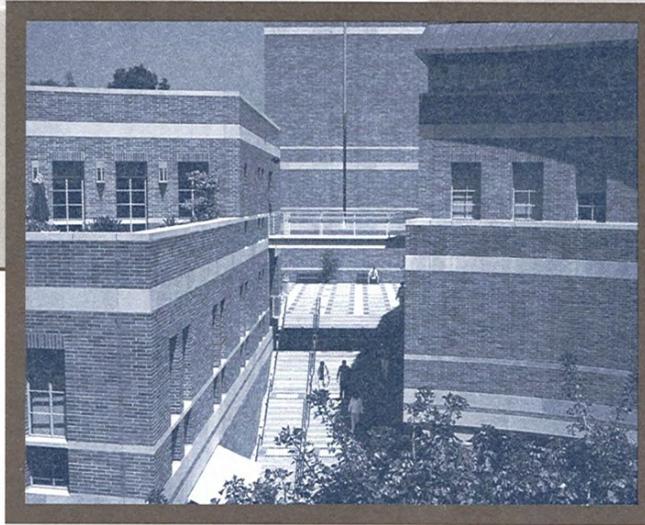


UCLAAnderson
School of Management



**LOCAL ASSIMILATION OF AN ENTERPRISE
SYSTEM: SITUATED LEARNING AT A DISTANCE**

YUTAKA YAMAUCHI

PALO ALTO RESEARCH CENTER
3333 COYOTE HILL ROAD | PALO ALTO, CA 94304

E. BURTON SWANSON

UCLA ANDERSON SCHOOL OF MANAGEMENT
110 WESTWOOD PLAZA | LOS ANGELES, CA 90095

SEPTEMBER 13, 2009

INFORMATION SYSTEMS | WORKING PAPER 1-09

INFORMATION SYSTEMS WORKING PAPER SERIES

The IS Working Paper series is a publication of the Information Systems Research Program (ISRP). It provides for the early dissemination of research by the IS faculty, students, and visitors, usually prior to its more formal publication elsewhere. The IS Reprint series includes these more formal publications, many of which supersede the original working papers. To obtain a downloadable index to both series, please visit our website at:

<http://www.anderson.ucla.edu/documents/areas/fac/isrp/wp-index.pdf>

Comments and feedback on our working papers are welcome and should be directed to the authors. Because their authors typically revise working papers within months of their issuance, we maintain only the most recent three years of the series for distribution. Most of these are downloadable from our website (see above). For copies of older papers, please contact the authors directly.

Publication of the IS Working Paper Series is made possible in part through the generous support of the ISRP by the IS Associates. For further information on the IS Associates, please visit their website at:

<http://www.anderson.ucla.edu/x574.xml>

**LOCAL ASSIMILATION OF AN ENTERPRISE SYSTEM: SITUATED
LEARNING AT A DISTANCE**

Yutaka Yamauchi¹

Palo Alto Research Center

3333 Coyote Hill Road

Palo Alto, CA 94304

yutaka.yamauchi@parc.com

E. Burton Swanson

UCLA Anderson School of Management

101 Westwood Plaza

Los Angeles, CA 90095

burt.swanson@anderson.ucla.edu

Revised September 13, 2009

¹ Corresponding author.

LOCAL ASSIMILATION OF AN ENTERPRISE SYSTEM: SITUATED LEARNING AT A DISTANCE

ABSTRACT

Once a new information system is introduced, individuals struggle to make sense of it. Over time, it must be somehow assimilated into everyday work practices. Enterprise systems, because they are complex and integrate work across functions and distance, pose special challenges to learning at the firm's periphery, where local users are distanced from both the centralized system and others elsewhere. The present study, using direct observations and interviews at a bank in which a new CRM system was introduced, explicates the local learning process. Findings suggest that in assimilating the system, bank representatives created pockets of familiarity within which they routinely worked with it and outside of which they competently ignored it. Even within familiarity pockets, routine use of the system masked much that was not known. In short, in local assimilation of enterprise systems, knowing in practice may be constituted from what can be competently ignored by users.

Keywords:

Enterprise systems, Customer relationship management (CRM), Technology assimilation, Situated learning, Work practices, Familiarity pockets

LOCAL ASSIMILATION OF AN ENTERPRISE SYSTEM: SITUATED LEARNING AT A DISTANCE

1. Introduction

As is well known, when a firm first introduces an information system, employees do not simply begin using it. Rather they are often overwhelmed by the complexity (Brown & Newman, 1985), unsure as to where the information they provide goes (Boudreau & Robey, 2005), and even worried about breaking something (Carroll & Mazur, 1986). Still, most will eventually become so competent as users in their everyday work, that they put their hands to the system without much thinking about it. This learning process has been termed “assimilation” (Armstrong & Sambamurthy, 1999; Cooper & Zmud, 1990; Fichman & Kemerer, 1997; Gattiker & Goodhue, 2005; Huigang, et al., 2007; Meyer & Goes, 1988; Purvis, et al., 2001; Swanson, 2004) to suggest that the system and its use must eventually be fused into everyday work practices. But how is such learning accomplished, and what is it that is actually learned, in particular for systems as complex as enterprise systems?

Enterprise systems, such as Enterprise Resource Planning (ERP) and Customer Relationship Management (CRM) systems, aim to support a whole enterprise, typically by means of purchased packaged solutions (e.g., Davenport, 1998; Rigby, et al., 2002). By their complex nature, they pose significant challenges for learning. First, a firm must configure its packaged solution by selecting from an enormous number of optional features such that business processes can be properly coordinated across multiple functions and locations. Second, the firm must take an implementation approach that accomplishes an often complex and precarious transition to what amounts to a revamped work system

(Robey, et al., 2002). Third, the firm and its workers must over time and across functions and locations gain familiarity, knowledge and new capabilities from use and assimilation of the system such that enterprise value is actually achieved. Interestingly, this third enterprise system challenge, in contrast to the first two, has to date received comparatively little attention from information systems researchers.

Nor have practitioners been fully cognizant of the assimilation challenge. Because enterprise system packages are commonly marketed as “solutions” in themselves, and because the software itself dictates much of the interaction with its users, some adopters of these systems may see the challenge in overly simple terms, where users must simply grasp new set procedures. More sophisticated adopters will recognize that a new work system must be built around use of the enterprise software and that such a learning process will be substantially social, where users will learn much from each other in creatively fashioning their new routines. Indeed, substantial research suggests that learning in work systems is greatly facilitated through a specialized “community of practice” (Brown & Duguid, 1991; Lave & Wenger, 1991; Wenger, 1998). But this in turn suggests a deep puzzle at the heart of enterprise system assimilation. Because these systems are intended to coordinate work across the firm, they are typically implemented to reach all of its diverse and far-flung units, many of them small, where supportive community may not only be different from that elsewhere (see, e.g., Rolland & Monteiro, 2002), but also thin or even lacking altogether. In such circumstances, how is the needed organizational learning accomplished? Operating at distances both from the central system and from each other, how do local units cope and learn, and eventually assimilate the enterprise system, with what broader consequences?

In the research reported here, we sought basic insight into this puzzle of enterprise system assimilation, through a case study focused at the local level, drawing on detailed observational and interview data collected in related natural settings, paying close attention to situated practices. Specifically, we studied one bank's representatives and their use of several systems including a newly introduced CRM system in several local branch offices. We interviewed various representatives to uncover how they encountered the systems, began to learn how to use them, experienced difficulties, and in some cases eventually came to use them in a routine manner. We followed the interviews with observations of the reps actually working, i.e. interacting with customers, servicing accounts, and using the systems. We observed not only routine use but also various troubles encountered and the ways in which reps dealt with these. From these interviews and observations, we arrived at several insights into the local assimilation process, including the role played by what we term the user's "familiarity pocket" within which he or she locally engages the larger global system, while otherwise ignoring most of its aspects. We discovered how situated learning in relatively isolated local offices could take place at significant physical and social distance.

While our findings by no means resolve the broader learning puzzle, they do illuminate the local challenge. They contribute too to the nascent literature on learning in networks of practice where multiple communities prevail, as in the typical firm (Brown & Duguid, 2001). While research has begun to explore the problem of learning across communities (see, e.g., Pan & Leidner, 2003), it has given less attention to learning where community may be thin, or where work is coordinated by a distant centralized system. Whelan (2007), examining electronic networks of practice, concludes that relatively little is

known about the dynamics of knowledge exchange in such networks. The research reported here examines assimilation of new centralized technology at the periphery of the firm. We find that in the case of working with enterprise systems, local learners must navigate a territory which for most will remain largely a realm of the unknown, even as they gain a certain competence, while sharing knowledge only thinly across localities. Local assimilation of enterprise systems apparently entails learning to live with something that remains at something of a distance, unrevealed in its broader aspects, even as comfort, if not mastery, is gained in its presence. Thus, relative to other recent research, we describe a rather different kind of situated learning and its ramifications.

In the balance of the article, we provide further background, describe the study we undertook, and present our findings and insights, together with their implications.

2. The Local Assimilation Problem

While organizational assimilation of any new IT can be problematic, the assimilation of enterprise systems presents special challenges, deriving from the centralized and integrative nature of the systems, and the fact that they are to be used locally among the firm's diverse and widespread units, where situated learning necessarily takes place. We elaborate on these special challenges next.

2.1 Enterprise Systems

When firms adopt and implement enterprise systems, they undertake projects of remarkable scale and complexity, as is well known (Davenport, 1998; Markus & Tanis, 2000). The packaged software itself contains a myriad of functions from which the firm selects a subset to comprise its configured solution. A centralized business logic underpins

this solution, as the purpose of the system is to coordinate business processes across functions and locations. The needs of individual units are accordingly subordinated to the larger purpose. A local unit may have relatively little input into the configuration decisions. The functionality it receives is likely to be both more and less than it might have wanted.

Too, when the firm implements the enterprise system, deploying it across units, it usually does so in stages which reflect a central plan. (In the “big bang” approach, it deploys the system all at once across units.) Whether a unit receives the system relatively early or late relative to others, with all that implies for its own adaptation and learning, it must accommodate itself to the broader organizational interest. A local unit may have relatively little input into the overall implementation schedule. It may also have little say as to the advance preparation and training to be undertaken.

As a consequence, when an enterprise system is received by a local unit, it frequently arrives as a very large and complex system to which the unit must adapt its work processes and fit them to those of others within an imposed time frame. The local unit may or may not see its own practices as benefiting from the new system, as the overall purpose is support of the enterprise as a whole. The local unit which previously had substantial flexibility in determining its own practices may now find itself required to provide new information to others, in particular, to enable more effective coordination across units. Too, the local unit may be expected to capitalize on new information made available to it, as with CRM systems that provide customer service personnel with a more complete view of the customer through a centralized data base.

Under these circumstances, it is not surprising that many implementations of enterprise systems have foundered, met resistance and even been abandoned (Davenport, 1998; Scott & Vessey, 2002). Still, many more others have apparently been successfully accomplished, and enterprise systems are now widely deployed among firms. What is less clear is the value firms have gained from their implementations. Research suggests that upon initial implementation, firm performance often drops, rather than improves, as units grapple with the transition (Markus & Tanis, 2000; Ross, 1998). Firms often underestimate the amount of organizational learning needed to assimilate the enterprise system. They may underestimate in particular the problems of local assimilation, which we address here.

2.2 Local assimilation

Situated learning theory (Brown & Duguid, 1991; Lave & Wenger, 1991) suggests that assimilation of new technology by those that use it is fundamentally local. That is, users will learn primarily through their locally situated practice and secondarily from their interactions with those located elsewhere, engaged in other practices. The theory suggests in particular that learning hinges on local communities of practice, where workers learn from each other in executing the tasks themselves. The extent to which the spatial reach of these communities can be extended through electronic communications is a subject of intense interest to researchers (Roberts, 2006). Substantial research supports the importance of community in facilitating organizational learning (Brown & Duguid, 1991; Wenger, 1998).

However, many users of enterprise systems will work in small units that are distant both from the firm's headquarters and from each other. Many will work at the firm's

periphery, interacting with customers or suppliers. These users must get their own daily work done and face various local exigencies. For those that interact with customers, for instance, these interactions will likely have priority over those with a computerized system. For those who interact with suppliers, the situation is similar. For such users of enterprise systems, carrying out the local work at hand will dominate whatever use they make of the systems. On the whole, then, remote users are likely to shape and interpret their interactions with enterprise systems according to their local practices, and to adapt these same practices to the systems only insofar as these practices can be bettered from their own local perspectives.

Still, enterprise systems will impose certain requirements on local work practices (Boudreau & Robey, 2005; Rigby, et al., 2002; Robey, et al., 2002; Soh, et al., 2000; Soh, et al., 2003). In particular, because these systems are intended to integrate work across units of the firm (Gattiker & Goodhue, 2005; Kallinikos, 2004; Ranganathan & Brown, 2006), users may be called upon to interact with a broader range of users located elsewhere. These interactions may also now be more formalized and automated through the system itself. For instance, users who formerly called their managers and colleagues on the phone to resolve work issues, may now be expected to enter these issues into the system for tracking purposes (Sachs, 1995). Management may also seek to standardize work practices across locations (Rolland & Monteiro, 2002).

Regardless, local users of enterprise systems may find that their interactions with others in different functions and locations do not come so easily. They may never personally meet the others with whom their work is to be coordinated. They may not even know who these others are and how they themselves use the system. At their own remote

locations, users may also be just a few doing the same work, furthering their isolation. Thus, while tight coordination across organizational units may be sought, communication and community in the use of enterprise systems may be thin.

2.3 Meeting the challenges

Notwithstanding these challenges, enterprise systems are in most situations eventually assimilated into everyday work practices. Substantial prior research, in particular that which takes a practice perspective (e.g., Brown & Duguid, 1991; Orlikowski, 2000; Suchman, 1987) suggests several ways in which users learn about new IT and come to assimilate it. We can expect to observe this same learning in the firm's local assimilation of enterprise systems but also specific instantiation of it in light of the challenges at the local unit.

First, users of newly deployed IT typically learn about the technology by experimenting or even playing around with it. Even after training, in carrying out their work, they initially engage in substantial trial-and-error activity (Swanson, 2004). In this way, they discover how to effectively interact with a new and unfamiliar system. Trial-and-error learning also enables users to interpret and collectively make sense of a new technology (Edmondson, et al., 2001). It is, however, not clear how local users who face exigencies of local work can actually experiment with the complex enterprise-wide system. Users cannot possibly try all of the vast number of features and make sense of each. Moreover, it is not clear how users can learn how to work with other users at a distance through trial and error. Even through such trial-and-error learning, they may not be able to reach any clear mutual understanding with these other users.

Second, users of newly deployed IT often find what they perceive as mismatches between the technology and the work they must accomplish. Where the system lacks pliancy, and is not easily adapted, users will often create their own “work-arounds” (Gasser, 1986; Koopman, et al., 2003; Sachs, 1995). They then build these into their own preferred practices. In some case, users may seek to work around the new system, so as to continue practices associated with the legacy system to be replaced, a reflection of their inertia (Boudreau & Robey, 2005). These work-arounds are particularly important for local users who cannot easily comprehend how the complex enterprise system works and how others at a distance use the system. They need to find their own ways to use the enterprise-wide system in the local unit.

Third, users often discover new and unanticipated ways to make their own use of new IT (Ciborra & Lanzara, 1994; DeSanctis & Poole, 1994; Orlikowski, 2000). Some users can be quite creative in finding “features” and in appropriating the technology to accommodate or even advance their own practices. They are sometimes said to “reinvent” the technology when they enact it in this way. Researchers taking a human agency perspective have been much impressed with this phenomenon (see Boudreau & Robey, 2005). This reinvention is not done in a larger scale, but confined in each isolated local unit. Such local reinvention may be small and mundane. We need to understand what role these creative activities play in local assimilation.

Fourth, users may nevertheless rather quickly establish their own routines for working with new IT. In doing so, they may close down the “window of opportunity” for learning about a new system’s many features, leaving these under-utilized (Robey, et al., 2000; Tyre & Orlikowski, 1994). Routinization is, however, necessary to assimilation, as it

allows attention to the system to be subordinated to attention to the work (Swanson, 2004). Too, it reinforces and allows for further incremental learning (Feldman & Pentland, 2003). We need to understand how this routinization develops at a local unit.

In the local assimilation of a new enterprise system, we should expect to observe all these ways of learning. The relative few studies of the use of enterprise systems suggest as much (Boudreau & Robey, 2005; Wagner & Newell, 2006). It has not been made clear, however, how an enterprise system becomes assimilated in local units where users face particularly significant challenges. As will be seen, these findings from prior research help explain local assimilation. But we found too that what users learn in using and assimilating an enterprise system differs from what might be expected.

3. Research Design

Using a grounded approach, we undertook a “revelatory” case study in which a new enterprise system was introduced by a firm across multiple sites (Yin, 1984). Our subject was a fast-growing bank that had recently implemented a packaged CRM solution provided by a leading vendor. We first interviewed several senior managers, including one person who oversaw CRM implementations in the parent company, and another who led the CRM initiative in the bank. We learned that the strategic vision for CRM entailed a “unified approach for managing customer data” in support of the bank’s growth. The new system was targeted in particular for use in the bank’s call centers and local branch offices, where contact with the customer was direct.

We concluded that local offices where customers come and interact with representatives would be promising to study because we could observe multiple sites

similar in almost all aspects. We were introduced to a manager who supervised the entire local office operations. In discussion with him, we identified five offices as potential sites, mixing those where CRM was used heavily with ones where it was not – hoping that we could probe variations in assimilation. We then talked with two field managers who supervised the five locations, to explain our study. With their approval, we carried out the study at the five sites over about a one-month period, making nine separate visits totaling more than 26 hours.

The five offices studied were identically organized. Each was staffed primarily by a pair of representatives. One was senior and licensed to sell financial products such as stocks, bonds and annuities. He or she had 16 years of banking experience on average. The other, a junior rep, typically was not licensed, but had substantial industry experience, more than 20 years on average, and was thus not in an early career stage. Notwithstanding this experience, the bank studied is relatively new and most of the reps were newly employed within the last few years.

Our planned study consisted of repeated fieldwork at the five locations. To capture the embodied, situated practices, we chose to observe the work closely in the work setting. Because of an unexpected policy change within the firm, we could not complete the planned longitudinal fieldwork—resulting in two to three days of visits per site. Still, we were able to obtain rich observational data on use of the system. Too, by studying multiple sites we observed varied ways and levels of using the same technology during the one and a half month period in the spring of 2004.

At each office, we began with an extensive interview with both representatives. We asked about their backgrounds in banking, in this particular bank and in using IT systems.

We also asked as to the training they received, the difficulties they faced in using the CRM and other bank systems, and how they learned how to use them. The ethnographic interview method was used to obtain concrete details of recalled events rather than abstract thoughts and opinions (Spradley, 1979). We took care to obtain what actually had happened in each specific situation described. While personal opinions and thoughts are important data, some of which we use here, we were oriented more towards ethnographic data even from interviews. Audio recordings were made and all were transcribed. We do not simply accept what participants say as somehow objective and correct accounts of the situation but treat the accounts as part of the situation itself. For this reason, we analyzed what participants said carefully, trying to understand how and why they said what they said and how their understanding manifested itself in both the speech and the situation described.

When the interview was concluded, we undertook work observations. With their cooperation, we shadowed the representatives by sitting or standing next to or behind them. Typically, reps began by showing us how the system worked and how they typically used it. Customers could of course walk in at any time. Because of obvious privacy and security concerns, we could not audio record interactions between the reps and their customers. Nor could we obtain any documents. Yet we could jot down various things observed in a notebook while excluding personal information. Detailed vignettes could be created from these jottings. Where possible, we asked the representatives questions to clarify and confirm our understandings.

Because of the similar arrangement across local offices, our analysis made use of comparison of the same aspects of system use and learning across multiple sites. If reps in one site used a particular system feature in a creative way, we could see whether reps in

other offices did the same. We could even ask reps why they were not using a certain feature or using it in a different way from others. Such comparison enabled us to see the fuller picture as opposed to noting only what happens in a particular office. In fact, comparison confirmed much similarity in the reps' work across sites. We could then inquire into why reps did certain things by revisiting the background and historical data obtained from interviews. From this analysis, one particular theme emerged in interpreting our findings, the notion of "pockets of familiarity." Subsequently, we revisited the data to develop a clearer picture of this notion, confirming some of its aspects while also augmenting it. The findings provided below are developed around this notion, delineating its various aspects. We provide evidence, largely qualitative, for each of our claims and entertain alternative explanations.

4. Work of the Bank Representatives

At each local office, each representative has his or her own personal computer, connected to the corporate network, and loaded with or with access to four main applications. One is the bank account management system (hereafter Bank System) with which reps can open accounts, service accounts, look up activity history, and so on. All the account data are managed in this system although reps in local offices have limited access to them. The second system is the CRM system, which records all data regarding each customer so as to provide a "360 degree view." After a customer contacts a call center for some matter, he or she may walk into a local office and ask for an update. The CRM system can help reps to understand what has been done and thus provide timely and accurate service. Another system is Lotus Notes, which reps use mainly for email. They typically

receive a few messages each day from their manager, who supervises several offices and works in the region's central office. The fourth application is a Daily Activity Report compiled within Microsoft Excel files. All newly opened accounts are listed in the Excel file, which is saved on a network drive accessible to the bank's managers. Management and marketing people can thus monitor how many accounts are opened when and where, so that they can create an effective marketing strategy.

The representatives' work has a similar pattern across the offices. Reps serve customers who walk in to ask questions, open an account or buy financial goods. Typically, two to six new customers open an account each day at each office. Reps also service existing accounts. For example, they follow up on newly opened accounts to see whether the accounts are properly funded after deposits are made. Existing customers also call in to inquire into their balances or the status of a requested service.

When a new customer walks in, one of the reps greets him or her and initiates the usual interaction, inquiring into what the customer wants to accomplish by the visit. If the customer wants to open an account, the rep solicits specific information such as the duration, amount, etc. and also gives the customer an application form. The rep makes copies of necessary documents and helps the customer prepare the form. Then, the rep logs onto the Bank System to open the account, and copies information from the form into the system. Lastly, the rep prints out a shipping card to ship necessary documents and a check to fund the account to the deposit center.

After an account is opened and the customer leaves, the rep enters information about the interaction into the CRM system. He or she opens a page for "communication records" and starts filling in the fields. Typically, the rep types the last and first names of the

customer, an activity category such as opening an account, a marketing code such as local newspaper, and some comments. Even if the interaction doesn't entail opening an account, the rep needs to record all interactions with those who walk or call in, in communication records. Where the rep receives a phone call from someone who does not identify him or herself, casually inquiring into rates, the rep identifies the caller as "unknown" in the record. After opening an account, the rep also needs to fill in the Daily Activity Report on Excel.

To send special requests to the back office, for instance, issuing a check in case of the account holder's death and changing the account type, the reps fill out service requests on the CRM system. Reps need to indicate the account number and name, who to assign the request to, the type of the request, a brief explanation, and so on. The CRM system enables the reps to keep track of service requests they have sent out as well as the ones they are assigned to.

5. Learning Processes

We draw on selected data to describe the learning process from initial use of the system to routine use. We found varied but consistent stories across the situations examined. We report exceptions, deviations and unique cases where relevant. We introduce the concept of a user's pocket of familiarity informally, by example, reserving broader theoretical discussion for later sections.

5.1 Gaining Initial Familiarity

When new employees joined the bank, they were sent to a training center for one week, where they were taught the job. In half of one day, the new reps learned both the

Bank System and the CRM system. Typically only an hour or so was spent on the CRM system. Reps were introduced to basic tasks such as opening an account and servicing it. Instructors lectured and had participants do some hands-on exercises. Several reps complained that they were given too little training. After this training, they were sent back to their own offices and told to start working using the system.

Several reps we asked said that back in their office they could gain a good sense of the system by “playing around with” it for a while, i.e. using it while not facing a customer in an actual business situation. In this way, the rep explored the functionality of the system apart from actual work. Additionally, several reps said they were “walked through” the system by a colleague, which was clearly helpful to their gaining familiarity with the technology. One said:

I didn't feel that I got enough training on Service Request to know what I was doing, once we had to do it, but once I was walked through it, if I did a couple of them it was fine. They're very easy programs.

We specifically asked about rep experiences in using the systems in their work for the first time. Reps described an initial period of confusion when faced with the systems. Although they had already been introduced to them, when facing them within the work environment, they needed to re-familiarize themselves. Particularly, the CRM system was very complicated because there were many ways to do the same thing and various unused functions were available without being disabled.

It was brand new for me. It can be **confusing**, this is not the easiest system in the world. **There's many doors, there's many**

windows, as that's how I put it, to go through to get to accomplish a certain thing, but once you learn how to do it, it's easy, **there's not a lot to it.** [emphases added]

The initial encounter was “confusing” for this rep. There were “many doors and windows” to do a certain thing. Nonetheless, it is important to note that she and others said it was “easy” to use the system after going through several instances of use. She said, “There’s not a lot to it,” despite the system’s many doors and windows. This contrast between “many doors and windows” and “not a lot to it” reveals an interesting aspect of this learning.

Another rep said a similar thing.

As far as the CRM System, it was **a little confusing** at first. Because I **didn't really know what the tabs meant.** I mean, I had some training on it, but I wasn't really sure. (omitted) So I would click on all the tabs and just kind of **get a feel.** And just kind of, I guess, **play around with it,** just to understand what exactly it entailed. And even now **there's just a few things on there that I'm not too familiar with,** but basically, whatever functions I need on the CRM System, **I know how to use.** (omitted) As I started using it, it was just kind of like it kinda **flowed,** began to flow. But now **it's really easy to use.**

Again, the system was “a little confusing” as she “did not know what the tabs meant.” In contrast, “it’s really easy to use” after “getting a feel.” However, we found that this “easy to use” did not necessarily mean that reps had learned that much about the system. While observing, we asked reps questions about various parts of the systems. We were

surprised to find that they often could not explain what the parts were and how to use them. This was so even for parts such as text fields and buttons within the “tabs” they readily used. They could not explain what some fields and menus on the tabs were. Thus, when reps say, “There’s not a lot to it,” it does not necessarily mean that they learned most of the things they were initially confused about. Still, they were able to overcome the confusion and perceived the system as if it “flowed.”

The findings suggest that users familiarized themselves with the system not by learning most parts of the system but by learning a few parts and ignoring the rest. We call what is learned a “pocket of familiarity,” implying that users create a pocket within which they competently act on their knowledge, while competently ignoring that which lies outside it. A user’s competence is thus two-fold. We explore this notion further below.

5.2 Coping with Troubles

We found that simply following a procedure did not work in general and troubles were necessary to learning. Using a system required knowing typical troubles. For example, most reps mentioned the same difficulty with a query function, for locating a particular customer in the database. In fact, management was frustrated because reps didn’t find existing customers and created new entries, resulting in duplicates. Similarly, a particular function “Service Requests” was quite complicated. When we asked about it, all the reps said that the service requests were confusing. There were little things that users needed to know. For instance, when reps went to the “Service” tab to create a new request, they found the form already filled in – one said it was a “dummy account.” They did not know where this data came from. Instead of changing this form, they clicked the “New” button to open

up a blank form. Similarly, a service request was left unchanged even after the submit button was pushed, causing confusion as to whether it was really sent. Some reps returned to the original screen to double check, closing the request screen each time. The same trouble happened with the "Communication Record," which remained on the screen even when the rep pressed the submit button. These issues were not difficult once known. But initially, the rep wonders what was done wrong.

A more important problem is that reps are often not sure whether a service request has been acted upon, even if they made sure it was submitted. One said, "You send it off and then you wonder what happens to it." That is to say, even if a rep saw a service request "closed," the request had not necessarily been taken care of as intended by the back office. This sometimes causes trouble because a customer may not receive an interest check, an address may not be updated, and so on.

One representative described how she identified the problem and tried to solve it.

The service request **is closed but it's not done**. (omitted) I can see who it was assigned to, well what did they do? And then I look at the notes, well a lot of times they'll say completed and that's it, well when I look at the account, it's not completed, so then I have to go back in, reopen it, (omitted) and I'll put a little note saying, "I opened it because when I check the account, I notice that it has not been completed, please complete."

Here, the rep was trying to negotiate the ways she and others should work together with the system. She did try to do so within the system, writing on the note field. But then, she could not resolve this issue and saw similar problems multiple times.

And when I've taken it upon myself and I've called the people I've assigned and I say, "So what are you doing? Why did you close this?" She says, "Oh well I saw it and I know I had to do it, so then I closed it." So she said she was thinking of it as a communication record, and I said, "Well no, when it's a service request, you keep it open until it's done."

This rep did not know what the people in the back office understood and meant by closing the request. She found, according to her, that the person in the back office understood it as a communication record, for which closing does not have any special meaning. This experience shows how the rep discovered the problem and tried to resolve it. She had a certain expectation as to what others should do – if a service request is closed, it should mean it is completed. As she repeatedly observed that the reality betrayed her expectation, she investigated. She then used, first, the system feature to resolve the issue with a note field and, second, phoned to directly clarify with the person involved. We were surprised to see that even with such clarifications, the rep did not know confidently that the back office handled the requests properly. As a result, she double-checked to see if the requests were handled by constantly looking at the account status on the Bank System; e.g., to see whether an interest check had been cut, she could check whether the interest amount was withdrawn. Here, we see that reps' understanding is not necessarily clear, even after they encountered the trouble and inquired into it. They did not trust the people in the back office even when they clarified the issue by talking to them. Two reps even said that they specifically tried to avoid using service requests as much as possible, by trying to resolve issues over the phone.

The one rep then continued,

It's a lack of training or you know **maybe I was trained wrong**, maybe they don't want us to keep them open, maybe they do, it could be just a difference of opinion as well. But in something like this that's a pretty big difference of opinion so, it needs to be, you know, everyone needs to be taught the same thing for something like this.

Note that the rep did not mean that she was right and people in the back office were wrong. She was only guessing for the practical purpose of getting the work done. Although many reps complained about parts of the systems and how others used them, nobody could explain to us clearly how exactly the system worked and how others were supposed to use it. Particularly, the behavior of the CRM system remained a mystery, as to how it handled service requests and communication records when the submit button was pressed, what the data populated in the forms when tabs were opened were, and so on. Nonetheless, reps could eventually competently use the system by building double-checking and workarounds into their pockets of familiarity.

There were other examples of implicit guessing and assuming. When one rep was using the filter function to obtain the service requests assigned to her or her colleague in the same office, using "My team's request," she had assumed that "My team" meant herself and her colleague. She found, however, that she had retrieved all the requests assigned to everybody. Another rep was confused about the "new" column of service requests listed in the Service tab. She had thought that "new" (as indicated by asterisks in the column) meant that the request was "open." When she found that some "new" ones were actually "closed,"

she became confused and tinkered around to learn more about it without ever successfully figuring it out. We also observed other examples of double-checking. For instance, whenever a rep created a new communication record, she went back to the “Home” tab and confirmed that the record was submitted and not left open. By this practice, she took control of the records as needed.

In their learning, reps thus made various implicit assumptions and guesses as to how the system and other users were interacting with them, incorporating these into their familiarity pockets. Through troubles and unexpected events, some of these assumptions were surfaced and reworked, resulting in a more elaborated and deeper familiarity. Yet, unclear understanding of the behaviors of the system and other users remained. That which was “known” within the familiarity pockets was accompanied by much that was unknown.

5.3 Asking for Help

When facing troubles, reps typically sought help. We observed reps asking colleagues questions and also teaching others. In one case, a customer asked over the phone a question about retirement accounts services. Not knowing the answer, the rep put the customer on hold for a few seconds and asked her nearby colleague a question. Told “no,” she then provided an answer to the customer. Another rep had learned the same thing in a different context, as in his first job with the bank four years earlier he was a back-up person in a call center. He was able to ask colleagues questions in the middle of calls by putting the customer on hold. Now, he found that, in a local office where a customer is sitting in front, it is difficult to put the customer “on hold.” Still, being able to question right in the middle

of having trouble is obviously an effective way to learn how to use a system. Such light-weight interaction between co-located peers can be supportive of learning.

Simply asking around is one way to seek help from colleagues. We found that reps tried to seek help from various sources. They contacted the training instructor who had said that he would be available on whatever questions they might have. They also contacted the regional operations assistant who centrally manages all service requests and is knowledgeable as he or she has been through a wide variety of work activities. Said one rep²:

Like if we had a question about something, like of the personal loan and line, (omitted) You usually can't get anybody on the phone so you might as well not try to call anybody. So you email, **I emailed Ince who's the trainer** pretty much for Bank and he's a pretty busy guy. He writes the manuals and he's done all the sales training and they have him going to different states. (omitted) So he always said, "If you have any questions just email me". So he's out of town and so then **I email like Ashley** and I try not to bug her because she's like the **assistant supervisor** and then she's always filling in at these different branches. She's got a lot of work on her plate so I emailed her and then she didn't know and so then she said she was going to **email Mandy**, who's her boss, which is **my boss** and I guess she's not feeling well. So meanwhile you don't, you know, you just have to wait I guess. (omitted) **I talked to Lea**, she doesn't know.

² All names are changed here and elsewhere in the paper, as are a few contextual details, to protect the privacy of our research participants.

In this particular case, the rep was tapping into various levels of support. She tried to contact the trainer who gave the formal training, the regional assistant, her manager and her colleague at a different location. Still she could not figure out the problem.

We observed several such instances of asking for help. In one, Sandra, a rep, needed to open a trust account of a special customer, a friend of the bank's top executive. Although the customer's wife was a trustee, she was in San Diego and not present as required. Sandra called her manager asking what to do. The manager called back and said that she should make an exception. Sandra then began opening the account on the Bank System. When she entered the wife's name as a trustee, it appeared before that of her husband. Sandra tried to fix this by deleting the wife entry and adding it back in, and also starting all over again. She printed the shipping card to see if the order would be corrected on the paper but found it the same. She then asked her colleague on site, but he didn't have an answer and actually had his own problem with his system. When she asked the colleague, "Do you have any suggestions, Elliot?," he replied, "I can't get on [UPS label printing page]." Sandra tried a few other things, becoming frustrated. She finally phoned Lea in another location, in an adjacent town: "Hey Lea, how are you doing?" She continued,

I'm doing uh trust for that one little big big customer (omitted).
(pause) And I put the wife's information (omitted) And it keeps
printing (emphasis) the wife's name first (emphasis)! (pause) Oh.
(pause) Oh. (long pause, listening) Yeah, yeah. It's Arlene with an
A. (pause) Dumb. (pause) Oh.

After hanging up, Sandra then remarked to her colleague, "Lea told me that it prints in the alphabetical order. Isn't that stupid." Again, here, we can see that Sandra used the

feature without anticipating trouble, implicitly assuming that the account owner name she typed in first should be printed first. She was already familiar with the system, but through such implicit assumptions. Through this interaction with Lea, she turned the implicit into the explicit.

It is important to note that while the rep relied on her colleagues for help with the trouble, she did not resolve it this way. What she found was that the order of names was by design. She discovered no way to print the names in the order wanted. Still, she gained some satisfaction and ceased the inquiry. She found that there was nothing she could do. She could blame the system as “stupid.” She could then claim that she was not doing anything wrong.

That colleagues doing the same work constantly query each other is hardly surprising. This contrasts starkly with the difficulty in interacting with members of other groups like the back office. Reps could rely on each other in the same office quite efficiently and in some cases on colleagues in another office. We observed no serious misunderstandings within the reps’ community of practice. In contrast, misunderstandings were not resolved, despite reps’ efforts, across different communities of practice. This inevitable difficulty faced in local assimilation needs to be overcome in different ways, i.e., double-checking.

5.4 Developing Personal Tools

Reps constantly invented new and unanticipated ways to better accomplish the work. For example, one representative improved her work by a small change in the process. When a new account is opened in the Bank System, the customer’s birth date is entered. The Bank System then calculates and shows the age of the customer. Routinely this rep writes down

the age on the top left margin of the paper application form so that when she later adds an entry to a Daily Activity Log, she can just copy the age rather than calculating it anew. This makes the work both more efficient and more reliable, avoiding calculation mistakes.

Another example is that one rep opened a communication record entry page as soon as she began talking with a customer, when she picked up a phone or a customer walked in. The system automatically filled in the beginning time of the communication, which the other reps typically jotted down on paper for later entry.

One particular rep made a series of inventions. When creating a communication record for an exchange with an existing customer, representatives were required to “query” the database and locate the customer’s information. Because the “last name” was the search field listed on top, most reps we observed typed in the last name and clicked the query button. Yet, there were many occasions where reps did not know how to spell the last name or what the last name was. Too, common last names could produce a long list of customers, from which a rep needed to find the first name. Problematically, the system was not designed to allow easy scrolling. It showed only ten customers in one page. This rep discovered a way to query by the customer’s first name in case the last name was common. While this is not a surprisingly innovative way to use the function, we saw many others simply use the last name and scroll through a number of pages – in one case, a rep gave up and selected the “unknown” category.

Right when we were observing this rep, she also discovered a way to query by a phone number. In many cases, she wrote down the number because customers called and she had to call them back. She could find the customer easily this way because phone numbers are unique. She was visibly proud of this discovery. She had also used the service

request function in a way that was not only unexpected, but also came to be discouraged. She had used the function to keep track of her to-do's by submitting service requests to herself and later going through these to manage her own work. She was subsequently told not to use the function that way because it was not designed for that purpose.

We found that a system's breakdown can be a good occasion to trigger sensemaking. This rep told us one episode. When she received an inquiry from an existing customer about his account, the Bank System was down. Account information was typically obtained from The Bank System. She proudly explained to us that she found a way to obtain certain account information within the CRM system, which downloaded data from The Bank System every night. No training had been provided as to how to tap into them. Inventions such as this were largely made out of necessity, either when the system was down or when system functionality was limited.

Thus, over time users develop personal tools so that they can make their work more efficient. These personal tools are largely small and mundane but instrumental for learning and expansion of the familiarity pocket. The personal tools often shared among colleagues. Learning is thus extended, often building on the routine. System use is incrementally tuned by the invention of personal tools. While we have already seen how reps deepen their familiarity pockets by surfacing and reworking their assumptions, here we see reps broaden their pockets by tapping into new features and inventing new ways of using already familiar features.

5.5 Routine Use

When assimilated into everyday work practices, the system becomes part of the web of resources that users employ without thinking much about it. This is so even when users extend their routines, and thus their learning, in applying them to new situations. We offer one example.

Eric and Rick worked in the same financial center. One day, Eric received a phone call from a customer who had opened an account the week before. The customer said that the check she had submitted for deposit needed to be held for a while because her other bank was holding another of her checks prior to cashing it. Eric said that he would try to stop the check. He asked Rick to call their manager Manny. Rick could not reach him, but immediately began to enter a service request. Eric then said that he was going to call Manny on his cell phone because he also had another thing to discuss. When reached, Manny told Eric to call a regional operations assistant Aster and ask her to call the deposit operations department. Eric informed Rick of this and started calling Aster. In the meantime, Rick sent the service request. In the phone call with Aster, Eric asked, “Has Rick sent you (the request)?” Aster acknowledged he had. After hanging up with Aster, Eric told Rick that it was now “out of our hands.”

This is a very typical example of an assimilated technology in complex use. Six people including the customer and the deposit operations department were involved and three communication channels (face-to-face between, electronic service requests, and phone calls) were intertwined. Rick began using the service request routinely. He knew what he needed to do. Rick and Eric knew what the other was doing even without explanation. Note also that Aster expected Eric or Rick to send a service request. Although

the question “Has Rick sent you?” lacked the object (i.e., a service request), it was taken for granted and understood by her. One key feature of this routine was that participants produced their actions knowing how the system and others would react. Therefore, they did not need to monitor what the system or others were doing or coordinate with explicit explanation.

The resulting “out of our hands” indicates that Eric knew that Aster would handle the request properly. In contrast to what reps experienced with people in the back office, this knowing is markedly special. Because Eric had interacted with her many times in the past and knew her well, he could trust her. The fact that they talked over the phone was also important because reps rarely talk to those in the back office. Eric’s knowing Aster, however, is not full knowledge because Eric was comfortable removing himself from what would happen next and let it out of his hands. Here, his knowing included not only knowing whom to talk to but also knowing what he did not have to know.

Finally it is important to note that the task faced was not necessarily a simple one. The problem dealt with (namely, putting a hold on a check already submitted for deposit) was not an everyday one. Rather, it was one of the many exceptional cases that might arise, but are not anticipated. Still, the problem was dealt with through subtle extension of familiar routine. Moreover, learning resulted, even while bounds on familiarity pockets were reinforced, as reflected in Eric’s comment that the problem was now “out of our hands.” Thus, when we say the use of the system was routine we do not mean that people are just repeating the same thing. Rather, use is routine because the familiarity pocket is sufficient in its breadth and depth to accommodate most new situations and needed interactions with the system and others.

6. Discussion

While the learning processes observed in the present case closely parallel those identified in previous research, the notion of a familiarity pocket enables us to tie the learning together and gain insight into what is actually learned, and importantly what is not learned, in local assimilation of a complex, centralized enterprise system. Having by means of the case introduced the notion of a familiarity pocket, here we elaborate more systematically and thoroughly.

6.1 The familiarity pocket

We define a familiarity pocket to hold bounded work routines developed through situated interactive use of an information system. Note that we refer here to familiarity not with the system apart from its users, but rather with the routines by which users interact with the system and each other. Such routines are organizational, as they incorporate “repetitive, recognizable patterns of interdependent actions, carried out by multiple actors” (Feldman & Pentland, 2003, p. 95), including as we interpret it, the system itself. Each user’s familiarity pocket is accordingly understood to be built and differentiated according to the interdependent actions engaged in. Where users share a routine, they are understood to do so through their interactions, and because each must interpret the actions of others, their respective familiarities with the routine will be different.

Thus, as situated learning theory would suggest, users through familiarity pockets come to understand and eventually assimilate a system not in isolation but in the social action of using it. Using the term “bounded routines,” we suggest that among the many possible ways of using a system, users form a limited—bounded—set of routines, which

are themselves related and interact with a similarly limited set of system features and functionality. The bounds are not imposed, but rather generated through the history of system use, giving shape to the familiarity pocket, and also reinforcing it.

Outside a user's familiarity pocket, then, is an unknown world, which, we emphasize, includes many aspects of the system. Indeed, in the case of enterprise systems, the vast majority of its features may remain unknown to an individual user. This unknown world, however, does not confuse or overwhelm the experienced user, because by means of routines he or she effectively knows to ignore it. This contrasts with the neophyte user who in the absence of routines, is easily overwhelmed by apparent system complexity. The experienced user's ignorance is an achieved one. He or she finds ways to overcome not-knowing, not so much by learning what is not known, but by learning how to competently ignore it.

Within a familiarity pocket, users thus know how to use the system, but not necessarily in anticipated ways, nor with the knowledge that might be expected. Users often do not fully understand even the parts of the system they routinely use. When asked, they cannot confidently explain how the system works. Rather, they readily make assumptions as to how it works, often unconsciously. When assumptions are confronted with counter-evidence, users simply work them out anew. Such indeterminateness did not seem to bother the reps much in their everyday work—they are familiar with using guesses—and they did not mention it unless we asked them about it. In short, the knowledge represented by a familiarity pocket pertains to the constructed routines and does not imply broader understanding of the system and its functionality.

The notion of situated interactive use is further important, as it underscores the reflexivity in the actions we observed in routine use. Where the system is assimilated, users typically interact easily with it, knowing how it should react to each of their own actions. Where the system is not yet assimilated, familiar interactions are lacking or break down. In such circumstances, users conduct small experiments, trying different ways of getting the work done, gaining in their familiarity through conscious effort. In this way, they build and reinforce their familiarity pockets, and their initial intensive trial-and-error usage substantially falls away.

6.2 Patch-work use

Still, we also found that even within their familiarity pockets, users did not always gain the knowledge that might be expected, but instead worked around their ignorance, devising ways to use the system reflexively without really understanding the interaction. For instance, when reps did not know whether the system had received submitted communication records as expected, they often double-checked by other means instead of figuring out how the system worked and gaining confidence in it. In effect, users incorporated what we term “patch-work use” of the system into their work routines and familiarity pockets. They patched over their ignorance of the system, more than they worked around an obvious short-coming of the system.

The same practice was observed in interaction with others located elsewhere. When they did not know whether others had acted on their service requests, reps often went to another system to double-check whether account information had been updated. They could have instead talked with these others and worked out a mutually understood way of using

the system, eliminating the need to constantly check on what had been done. Users distant from each other, however, can find it difficult to reach consensus on practices when they lack knowledge of each other's situations. They may find it easier to engage in patch-work use, simply by adding one more step to their own work processes. This does not mean that local users are happy doing such additional work. They are often frustrated. When we asked, they fumed about it. Still, patch-work use within familiarity pockets was itself routine.

6.3 Expanded familiarity

Earlier, we described how local reps coped with troubles, sought help, developed personal tools, and made routine use of the system studied. Through such learning processes, we suggest that a user's familiarity pocket can be expanded in terms of the situational breadth and depth of its routines. Situational breadth is gained according to the number and variety of situations within which the routines have been developed and executed. It is most commonly gained by extending or otherwise adapting existing routines to somewhat new situations. But we observed too that users in inventing their own personal tools, or being taught new tools developed by others, were able to gain situational breadth through applying these tools, incorporating them in new or adapted routines. For instance, when one rep discovered how to query by phone number in situations where other data were lacking, she added a new routine that incorporated the phone number field, which she had not used before.

Situational depth is gained according to the sophistication with which routines are invoked and engaged in particular situations, reflecting not only familiarity, but

understanding. We observed in particular that when a routine's implicit assumptions are brought to light through unexpected events, users can typically gain in their understandings. What was known only implicitly is opened to situated examination, interpretation, and validation. In this sense, familiarity pockets are deepened, although the knowledge gained remains imperfect and incomplete. Even when users explicitly inquire into a system's behavior, they often stop short of reaching clear understandings. Thus, the rep that discovered that the "new" (asterisk) column did not mean an "open" request, never did learn what it did mean. But learning what it did not mean enabled her to anticipate better how the system would respond to her actions.

6.4 Related cognitive processes

We note at this point in the discussion, that the familiarity pocket, as we define it, is not a psychological construct, but rather based in social and routine aspects of work practices. Still, it is interesting to consider briefly its relationship to cognitive processes. We note that cognitive scientists have proposed models by which new information is largely shaped by individual "knowledge structures" developed through previous experiences (e.g., Galambos, et al., 1986; Schank & Abelson, 1977). One of these is Hayes-Roth's (1977) developmental model of knowledge assembly, which explains how information is unitized in memory. It suggests that individual knowledge components that are initially separately strengthened, become schematized through gradually strengthened associations among them, and subsequently unitized into a single configuration that is invoked in its entirety. Such a process may well underlie the gaining of familiarity, where what is known as a routine is acted on more in terms of its entirety, than with conscious

regard for its various components. It may also help explain how users readily overcome complexities and competently ignore so many aspects of the systems they face. Here, however, we can only speculate about this.

6.5 Competent ignorance

Returning to our central questions, when local users are distanced from a highly complex enterprise system, how can they achieve competent situated practices of using it? At the same time, how can they coordinate as needed with remote others whom they do not know well and cannot interact with face-to-face? How is all of this made routine? The notion of a familiarity pocket helps us understand how this is accomplished.

What users achieve in building pockets of familiarity is not only a kind of knowledge-in-practice, but what might be termed a certain “competent ignorance.” Users learn first, to ignore that which lies outside their familiarity pockets. Within the pockets, they learn work routines that enable them to interact skillfully both with the system and with other users, but which further mask much which they do not need to know.

Whereas most prior research has stressed the knowledge that workers typically develop and display as users of systems, we suggest here that in assimilating new enterprise systems, local users may as much display an achieved competent ignorance. This counter-intuitive interpretation yields further insights into what we know from other research.

For instance, it is widely known that users are inclined to employ only a small set of a system’s features (e.g., Jaspersen, et al., 2005; Orlikowski, 2000). The achievement of a competent ignorance may underlie this phenomenon. Orlikowski (2000) remarks that “some properties provided by the artifact do not exist for us as part of our technology-in-

practice, while other properties are rich in detailed possibilities” (p. 408). As we interpret it, a user’s competent ignorance may enable certain properties “not to exist.” Importantly, users do not only engage in explicit sense-making and construction of coherent stories when facing systems that they do not know (Garfinkel, 1967; Suchman, 1987). Rather, users seek not just to make sense but to substantially ignore. Not-knowing is important in explaining not just situational breakdowns (Orr, 1996; Suchman, 1987) but also everyday routine execution.

The familiar concept of work-arounds can also be reinterpreted in part from the findings of the present research. As traditionally interpreted, a workaround is developed by a user to overcome misalignment between the system and the necessary work (Boudreau & Robey, 2005; Gasser, 1986; Koopman, et al., 2003; Robey, et al., 2002; Sachs, 1995; Wagner & Newell, 2006). The notion of patch-work use introduced here, while similar, is conceived rather differently. In patch-work use, users adjust their practices, not in response to misalignments, but in order to work around what they do not know. Working both within and around the system, they build patch-work use into their familiarity pockets as part of developing their competent ignorance. We speculate that such patch-work use may be rather more widespread than has thus far been observed, particularly in using complex enterprise systems, where familiarity pockets must embrace large unknowns. Certain of the work-arounds reported by Boudreau and Robey (2005), for instance, clearly involve patch-work use as we distinguish it.

We note too that while patch-work use of systems might seem inefficient, it can make coordination between distant users of enterprise systems robust. Where communications are difficult and trust in the actions of others is problematic through lack of knowledge, patch-

work use by means of double-checks and such can confirm that related work elsewhere has been carried out as expected. Patch-work use can thus enable a user's competent ignorance to extend beyond interactions with the system, to embrace those with others who are scarcely known.

6.6 Learning in assimilation

Finally, the present research sheds new light on the kind of learning that takes place in the local assimilation of enterprise systems. It suggests that the organizational learning that accompanies assimilation of these systems at the periphery of the firm may be rather different from that suggested by the literature that emphasizes the importance of knowing-in-practice to organizational capabilities and innovation (Cook & Brown, 1999; Orlikowski, 2002). We observe that much of the local use of enterprise systems is rather mundane and the knowing-in-practice achieved, while subtle and nuanced, can be of a very modest kind. In assimilating enterprise systems locally, users may not need to learn to work deeply with the technology, which may therefore remain at something of a distance. Rather, they may first of all need to build pockets of familiarity that enable a certain competent ignorance to be achieved in doing their daily work, where the primary focus may be not with the system, but elsewhere.

7. Conclusion

In sum, highly complex, large-scale global enterprise systems must be used across a firm, in diverse and widespread local settings. They must be assimilated not only at the core of the firm, but at its periphery, where employees interact with customers and suppliers. In the present study, we observed how one bank's CRM system was used and

assimilated by representatives in relatively isolated small branch offices. We found that the bank's reps came to use the system that was distant and complex by constructing pockets of familiarity that excluded or masked that which they did not need to know, enabling them to develop a kind of competent ignorance in their routine interactions with the system and with others. This finding suggests a caveat to the popular view of knowing as something of a pure, unalloyed competence, uncompromised in its acquisition. It suggests a more nuanced view of knowing that emphasizes not-knowing as an inherent part of knowing-in-practice. In local assimilation of enterprise systems, this not-knowing may explain how users accommodate themselves to the distances faced.

We acknowledge the obvious limitations of the study, which dealt with a single case of local situated use of an enterprise system, which we were able to observe only over a short period. We note that the system's users, while experienced in the industry, were also all relatively new to the firm and several were learning both the local work and the technology. In other, more mature work settings, the assimilation processes for newly introduced technology might be different, in particular where older legacy systems previously prevailed (see especially Boudreau & Robey, 2005). Too, assimilation processes may well be different in central offices, where management staff may make more sophisticated use of an enterprise system, e.g. seeking to mine its large database. Whether the notion of familiarity pockets has relevance in less routine work situations than those we observed may be questioned. More broadly, it remains for future research to probe the extent to which our insights have validity beyond the present case.

Looking ahead, the notion of familiarity pocket should be further developed and examined more closely, in terms of its assimilative role. Here we have only introduced the

concept; it remains to theorize it more specifically and probe its significance in the post-adoption learning process. Broadly, we need to know more about a pocket's characteristic size and growth pattern, and its interplay with assimilation. With regard to size, we suggest that it reflects users' need to balance what they know with what they don't know and can competently ignore in interacting with a system and its complexities. Size is important then not by itself, but in relation to the system and its greater potential. With regard to a pocket's growth pattern, it would be interesting to know not only its characteristic form overall, but the interplay between growth in breadth and depth, as the assimilative process unfolds. Too, we need to investigate alternative management and design interventions and how familiarity pockets may be purposefully shaped to advance organizational learning.

Most importantly, we should further explore the ways in which users' pockets of familiarity develop and adapt to each other collectively, through routine interactions. Broadly, to the extent that familiarity pockets overlap through interactions, communications should be facilitated, as community can be built. Thus, co-located representatives may be expected to adapt their familiarity pockets to each other, but less so to reps doing the same work at other locations, as we observed. Because reps had relatively little interaction with those in the back office doing very different work, their pockets no doubt overlapped very little with those of these others, and communications with this distant group were accordingly not only few, but problematic. While we were not able to directly observe familiarity pockets in the back office in the present study, our indirect evidence suggests likely misalignment with the pockets of the representatives working at a distance, at the bank's periphery. While the reps coped with this issue through patch-work use, it is possible to imagine a small design change that might better serve both parties to the

interaction. In sum, by investigating how different pockets of familiarity adapt or not to each other, we are likely to attain a deeper understanding of the ways in which collective learning and coordination can better be achieved.

From the practitioner perspective, managers and designers should understand that even where an enterprise system is deployed and put to use without much difficulty, the potential value of the system may hardly be tapped, particularly at the periphery of the firm. Distant users are likely to rely on a very limited set of routines with relatively few touch points to the system, where much is ignored, even within the routines themselves. Thus, where managers are concerned about infusing a new technology subsequent to its introduction, so as to better capture promised benefits (Jaspersen, et al., 2005), they might well focus on user familiarity pockets and how these might be shaped to advantage through facilitative interventions, such as providing new communication channels, e.g. electronic discussion groups, that encourage the sharing of best practices. In the present case studied, it seemed clear that learning and use of the CRM by the bank reps across locations might be advanced by such an intervention. It may be true in general, that managers underestimate the assimilation problems that centralized systems can raise at the firm's periphery.

More broadly, the notion of familiarity pocket gives managers and designers a way to focus on and evaluate user practices. In some cases, designers may wish to revisit assumptions about what users should be expected to know about systems. Some researchers have suggested that users need to take control of a system's behavior, which should by design be transparent to them (Brown & Newman, 1985; Norman, 1988, 2007). In the case of enterprise systems, however, this notion may be problematic, as it is not at all clear that local users seek such control and transparency. They may rather seek to relegate

as much as they can to background which they may then competently ignore. Their impulses may run counter to design assumptions. Designing systems to aid users in competently ignoring them for the most part presents an alternative challenge, one which egoless designers might wish to take up.

Both manager and designers can target familiarity pockets strategically. Managers seeking to encourage perceived ease of use and thereby gain wide support for a system, might aim for initially smaller pockets by focusing on the learning of a few basic tasks, while providing subsequent means, e.g. additional training, to expand upon these. Designers in turn might seek to facilitate faster growth of the pocket. Because users learn through troubles, designers might seek to build in equivalent means, e.g. challenging exercises, by which users “take the trouble” to advance their own work practice. Finally, managers should give increased attention to the multiplicity of groups and locations associated with use of their enterprise systems, and seek creative ways to facilitate troubleshooting and learning across these. Only then, by bringing familiarity pockets together through interaction, will the enterprise be likely to be integrated as envisioned by the system as conceived.

8. References

- Armstrong, C. P., & Sambamurthy, V. (1999). Information Technology Assimilation in Firms: The Influence of Senior Leadership and IT Infrastructures. *Information Systems Research*, 10(4), 304-327.
- Boudreau, M.-C., & Robey, D. (2005). Enacting Integrated Information Technology: A Human Agency Perspective. *Organization Science*, 16(1), 3-18.

- Brown, J. S., & Duguid, P. (1991). Organizational Learning and Communities-of-Practice: Toward a Unified View of Working, Learning, and Innovation. *Organization Science*, 2(1), 40-57.
- Brown, J. S., & Duguid, P. (2001). Knowledge and Organization: A Social-Practice Perspective. *Organization Science*, 12(2), 198-213.
- Brown, J. S., & Newman, S. (1985). Issues in Cognitive and Social Ergonomics: From Our House to Bauhaus. *Human-Computer Interaction*, 1(4), 359-391.
- Carroll, J. M., & Mazur, S. A. (1986). Lisalearning. *Computer*, 19(11), 35-49.
- Ciborra, C. U., & Lanzara, G. F. (1994). Formative Contexts and Information Technology: Understanding the Dynamics of Innovation in Organizations. *Accounting, Management and Information Technology*, 4(2), 61-86.
- Cook, S. D. N., & Brown, J. S. (1999). Bridging Epistemologies: The Generative Dance between Organizational Knowledge and Organizational Knowing. *Organization Science*, 10(4), 381.
- Cooper, R. B., & Zmud, R. W. (1990). Information Technology Implementation Research: A Technological Diffusion Approach. *Management Science*, 36(2), 123-139.
- Davenport, T. H. (1998). Putting the Enterprise into the Enterprise System. *Harvard Business Review*, 76(4), 121-131.
- DeSanctis, G., & Poole, M. S. (1994). Capturing the Complexity in Advanced Technology Use: Adaptive Structuration Theory. *Organization Science*, 5(2), 121-147.
- Edmondson, A. C., Bohmer, R. M., & Pisano, G. P. (2001). Disrupted Routines: Team Learning and New Technology Implementation in Hospitals. *Administrative Science Quarterly*, 46(4), 685-716.

- Feldman, M. S., & Pentland, B. T. (2003). Reconceptualizing Organizational Routines as a Source of Flexibility and Change. *Administrative Science Quarterly*, 48(1), 94-118.
- Fichman, R. G., & Kemerer, C. F. (1997). The Assimilation of Software Process Innovations: An Organizational Learning Perspective. *Management Science*, 43(10), 1345-1363.
- Galambos, J. A., Black, J. B., & Abelson, R. P. (1986). *Knowledge Structures*. Hillsdale, N.J.: Lawrence Erlbaum Associates.
- Garfinkel, H. (1967). *Studies in Ethnomethodology*. Englewood Cliffs, N.J.: Prentice-Hall.
- Gasser, L. (1986). The Integration of Computing and Routine Work. *ACM Transactions on Office Information Systems*, 4(3), 205-225.
- Gattiker, T. F., & Goodhue, D. L. (2005). What Happens after ERP Implementation: Understanding the Impact of Interdependence and Differentiation on Plant-Level Outcomes. *MIS Quarterly*, 29(3), 559-585.
- Hayes-Roth, B. (1977). Evolution of Cognitive Structures and Processes. *Psychological Review*, 84(3), 260-278.
- Huigang, L., Saraf, N., Qing, H., & Yajiong, X. (2007). Assimilation of Enterprise Systems: The Effect of Institutional Pressures and the Mediating Role of Top Management. *MIS Quarterly*, 31(1), 59-87.
- Jasperson, J., Carter, P. E., & Zmud, R. W. (2005). A Comprehensive Conceptualization of Post-Adoptive Behaviors Associated with Information Technology Enabled Work Systems. *MIS Quarterly*, 29(3), 525-557.

- Kallinikos, J. (2004). Deconstructing Information Packages: Organizational and Behavioural Implications of ERP Systems. *Information Technology and People*, 17(1), 8-30.
- Koopman, P., Koopman, P., & Hoffman, R. R. (2003). Work-Arounds, Make-Work, and Kludges. *Intelligent Systems, IEEE*, 18(6), 70-75.
- Lave, J., & Wenger, E. (1991). *Situated Learning: Legitimate Peripheral Participation*. Cambridge: Cambridge University Press.
- Markus, M. L., & Tanis, C. (2000). The Enterprise System Experience—from Adoption to Success. In R. W. Zmud (Ed.), *Framing the Domains of IT Management: : Glimpsing the Future through the Past* (pp. 173-207). Cincinnati, OH: Pinnaflex Educational Resources.
- Meyer, A. D., & Goes, J. B. (1988). Organizational Assimilation of Innovations: A Multilevel Contextual Analysis. *Academy of Management Journal*, 31(4), 897-923.
- Norman, D. A. (1988). *The Design of Everyday Things*. New York, NY: Doubleday.
- Norman, D. A. (2007). *The Design of Future Things*. New York: Basic Books.
- Orlikowski, W. J. (2000). Using Technology and Constituting Structures: A Practice Lens for Studying Technology in Organizations. *Organization Science*, 11(4), 404-428.
- Orlikowski, W. J. (2002). Knowing in Practice: Enacting a Collective Capability in Distributed Organizing. *Organization Science*, 13(3), 249-273.
- Orr, J. E. (1996). *Talking About Machines: An Ethnography of a Modern Job*. Ithaca NY: Cornell University Press.

- Pan, S. L., & Leidner, D. E. (2003). Bridging Communities of Practice with Information Technology in Pursuit of Global Knowledge Sharing. *The Journal of Strategic Information Systems*, 12(1), 71-88.
- Purvis, R. L., Sambamurthy, V., & Zmud, R. W. (2001). The Assimilation of Knowledge Platforms in Organizations: An Empirical Investigation. *Organization Science*, 12(2), 117-135.
- Ranganathan, C., & Brown, C. (2006). ERP Investments and the Market Value of Firms: Toward an Understanding of Influential ERP Project Variables. *Information Systems Research*, 17(2), 145-161.
- Rigby, D. K., Reichheld, F. F., & Schefter, P. (2002). Avoid the Four Perils of CRM. *Harvard Business Review*, 80(2), 101-109.
- Roberts, J. (2006). Limits to Communities of Practice. *Journal of Management Studies*, 43(3), 623-639.
- Robey, D., Boudreau, M.-C., & Rose, G. M. (2000). Information Technology and Organizational Learning: A Review and Assessment of Research. *Accounting, Management and Information Technologies*, 10(2), 125-155.
- Robey, D., Ross, J. W., & Boudreau, M.-C. (2002). Learning to Implement Enterprise Systems: An Exploratory Study of the Dialectics of Change. *Journal of Management Information Systems*, 19(1), 17-46.
- Rolland, K. H., & Monteiro, E. (2002). Balancing the Local and the Global in Infrastructural Information Systems. *The Information Society*, 18(2), 87-100.
- Ross, J. W. (1998). The ERP Revolution: Surviving through Thriving.

- Sachs, P. (1995). Transforming Work: Collaboration, Learning, and Design. *Communications of the ACM*, 38(9), 36-44.
- Schank, R. C., & Abelson, R. P. (1977). *Scripts, Plans, Goals, and Understanding : An Inquiry into Human Knowledge Structures*. Hillsdale, N.J. New York: L. Erlbaum Associates.
- Scott, J. E., & Vessey, I. (2002). Managing Risks in Enterprise Systems Implementations. *Communications of the ACM*, 45(4), 74-81.
- Soh, C., Kien, S. S., & Tay-Yap, J. (2000). Enterprise Resource Planning: Cultural Fits and Misfits: Is ERP a Universal Solution? *Communications of the ACM*, 43(4), 47-51.
- Soh, C., Sia, S. K., Boh, W. F., & Tang, M. (2003). Misalignments in ERP Implementation: A Dialectic Perspective. *International Journal of Human-Computer Interaction*, 16(1), 81-100.
- Spradley, J. P. (1979). *The Ethnographic Interview*. New York: Holt, Rinehart and Winston.
- Suchman, L. (1987). *Plans and Situated Actions: The Problem of Human Machine Communication*. Cambridge: Cambridge University Press.
- Swanson, E. B. (2004). How Is an IT Innovation Assimilated? In B. Fitzgerald & E. Wynn (Eds.), *IT Innovation for Adaptability and Competitiveness*. Dordrecht, Netherlands: Kluwer Academic Publishers.
- Tyre, M. J., & Orlikowski, W. J. (1994). Windows of Opportunity: Temporal Patterns of Technological Adaptation in Organizations. *Organization Science*, 5(1), 98-118.
- Wagner, E. L., & Newell, S. (2006). Repairing ERP. *Journal of Applied Behavioral Science*, 42(1), 40-57.

Wenger, E. (1998). *Communities of Practice. Learning Meaning and Identity*. New York, NY: Cambridge University Press.

Whelan, E. (2007). Exploring Knowledge Exchange in Electronic Networks of Practice. *Journal of Information Technology (Routledge, Ltd.)*, 22, 5-12.

Yin, R. K. (1984). *Case Study Research: Design and Methods*. Beverly Hills, CA: Sage.