Technology Analysis & Strategic Management
Vol. 15, No. 3, September 2003

Carfax Publishing
Taylor & Francis Group

Fitting Standard Software Packages to Non-standard Organizations: The 'Biography' of an Enterprise-wide System

NEIL POLLOCK, ROBIN WILLIAMS & ROB PROCTER

ABSTRACT *This paper investigates the development and implementation of a generic off-the-shelf computer package and the competing pressures for standardization and differentiation as this package is made to fit new organizational settings. The particular focus is on an Enterprise Resource Planning (ERP) system and its application within universities. In order for the ERP system to fit this setting a new module called 'Campus' is being developed. We followed the module as the current 'generic user' embodied in the software was translated to a more 'specific user' (a number of universities piloting the module) and back once again to a generic form of university user (the potential 'global university marketplace'). We develop the notion that these systems have a 'biography', which helps us to analyse the evolution of software along its life cycle and provides insights into the different dynamics at play as Campus is translated for use in a number of institutions and countries. The study draws on over three years of ethnographic research conducted in a British University and a major ERP Supplier.*

Introduction

After more than 30 years of software development for an ever-growing variety of institutional and organizational settings, few large-scale information systems are developed completely from scratch. Rather, most software applications are constructed by adapting existing 'packages' to new organizational contexts and settings. The category of systems that we are concerned with here—integrated enterprise-wide software solutions—are part of the software market that is bounded from 'below' (closer to the computer) by the market for operating systems, programming tools, and utilities and from 'above' (closer to the user) by end-user applications such as desktop productivity solutions and multimedia software. Despite their rapid diffusion across the widest range of sectors, however, it is increasingly evident that enterprise systems remain a costly and high-risk strategy. While suppliers aim to extend their solutions into as many different settings as possible, these systems do not translate easily across boundaries, whether between organizations within the same sector, between industrial sectors or between public and private sector organizational forms. Some of the consequences for those wishing to capitalize on the benefits of

An early version of this paper was first presented at the ACM Symposium on Applied Computing, Universidad Carlos III De Madrid, Madrid, Spain, 11–14 March 2002 and at the European Association for the Study of Science and Technology (EASST) conference, York, UK, August 2002.

The authors are at University of Edinburgh, Scotland, UK. Correspondence should be addressed to: Neil Pollock, School of Management, William Robertson Building, George Square, University of Edinburgh, Edinburgh, Scotland, UK (E-mail: neil.pollock@ed.ac.uk).

ISSN 0953-7325 print; 1465-3990 online/03/030317-16
DOI: 10.1080/09537320310001601504

packages is that they often undergo unwanted organizational change in adapting practices to models of work and organizational process embedded in the software. These dilemmas are particularly acute with integrated solutions (such as Enterprise Resource Planning (ERP) systems, the focus of this article), which seek new kinds of organizational flexibility and performance by capturing and integrating the full range of activities and transactions across an organization. Despite a growing literature on the uptake of enterprise-wide systems very little is known about how the gulf between standardized solutions and the specific contexts, practices and requirements of adopting organizations is reconciled both within user and supplier organizations. In the first part of the article we focus on what might be seen as an extreme example of the gulf, the application of ERP packages in universities. We discuss the various incommensurabilities that arise as a system is rolled out and adapted for use in one particular institution (a university we are calling 'Big Civic').

More generally, as concerns rise concerning the incommensurability of systems and contexts, there are demands for solutions that are already partially adapted to particular business settings (i.e. 'semi-generic' packages) and for increased user-involvement in the shaping of packages.[1] Alongside the adoption of these systems, then, there is an equally important story of innovation within supplier organizations and collaboration with package adopter as the technologies are adapted to these new contexts. In the second part we analyze the development of new ERP functionality, a university specific module called 'Campus', which is being built to facilitate the take-up of ERP by universities. The module is being designed around the needs of Big Civic and a number of other 'pilot sites' around the world and the eventual plan is to market Campus as a 'global university product'. We develop the notion that software packages have a 'biography', which helps us analyze the evolution of the software along its life cycle and provides us with insights into the different dynamics at play as Campus is translated for use in a number of institutions and countries.

The 'Biography' of an ERP System

ERP systems have become over a relatively short space of time key organizational technologies and suppliers have been highly successful in both creating and meeting this demand.[2] How can we begin to explain the success of these systems? We identify two important aspects, only the latter of which has been the subject of much detailed research. These are the related processes of 'genericification' and standardization that surround and accompany ERP systems. By genericification we point to the supplier strategy of taking a technology that has worked in one place and attempting to make it work elsewhere, and, in principle, 'everywhere'. This is evidenced by the fact that the software package we have studied, for example, was initially conceived for and used by manufacturing firms before being applied within non-manufacturing settings (pharmaceuticals, chemicals, retail, banking, etc.) and, more recently, non-commercial contexts (heath care, public sector, higher education and so on). Today, ERP systems are so widely diffused that they now commonly described as the *de facto* standard for the replacement of legacy systems in medium- and large-sized organizations, and it said that some companies find it impossible to work 'without one'.[3] The transferability of this software is possible because unlike conventional software development, packages are designed for a market and not a specific customer. Just how software is designed in this way is the subject of this paper and a theme to which we shall return.

In order for suppliers to reap the benefits of scale these systems must function in new settings in much the same way as they have functioned in all other settings. Several studies have considered the 'impact' such solutions have on organizations and how organizations often undergo expensive (and unwanted) organizational change and standardization in adapting organizational practices to process models embedded in the software, a topic about which much has been written. Davenport, for example, to quote the most cited author, discusses the case of a company with a well-established practice of giving its largest and most important customers preferential treatment, such as sending them products originally assigned to other customers. Once the new system was adopted, however, this was no longer possible as such a practice was proscribed by the technology.[4] The benefits of this strategy for suppliers are that they have to cater for only limited amounts of variation in product maintenance and new upgrades; for user organizations the benefits are having simple and guaranteed upgrade pathways. The downsides for suppliers are foregoing the high value-added markets for customized solutions; for user organizations the downsides stem from risk inherent in packages not adequately matching their requirements. It appears that in the conflict of interests, the supplier viewpoint generally holds sway: while marketing their systems as 'entirely flexible' and capable of coping with such idiosyncrasies, many suppliers actively encourage adopters to limit their attempts to tailor or modify the software by releasing upgrades and new software that are compatible only with the 'standard system'. According to a recent study of the German software house SAP, this proscription is not only limited to such direct methods but also permeates the whole ERP domain and its community of users. The systems embeds established ways of being used, as well as how the implementation is organized, all of which is reinforced textually through user documentation, by visits to other reference sites, and through the '... experience, competence and practices established in and shared by the SAP "development community" '.[5] There is, as one would expect however, no consensus about just how much (or little) customization can be carried out. Davenport also discusses the case of *Visio*, a small software company with unusual methods for accounting for its revenues and inventory, and how both these '... idiosyncrasies could be accommodated, but only with substantial extra programming'.[6] Light similarly points out that some organizations cannot completely adopt the standard model and therefore have no choice but to attempt customization.[7] Other, more ethnographic based research has attempted to similarly emphasize how these technologies are typically 'localized' by adopters. Wagner and Scott in their study of a US university describe how the standard templates in the ERP package were 'compromised' through 'skirmishes' and user resistance and this allowed the emergence of a much more 'local information system'.[8]

One body of literature, then, tends to emphasize how most adopters end up fitting their organization to the system (rather than the other way around), while another pays particular attention to the 'workarounds' and other strategies users deploy to adapt the technology to the specific setting.[9] A final strand has sought to reconcile these positions through emphasizing how technology and organization are often brought into alignment through a combination of quite complicated organizational change and software configuration, a process that is sometimes known as 'mutual adaptation'.[10] In general, ERP tends to be portrayed in the literature in one of two ways: either as potent 'global' technologies likely to transform all before them or as systems that work only because they are entirely 'domesticated' by their users. It is as if for some the universal applicability of ERP is beyond question, and for others there is no technology or standard able to work across many sites. Even those studies that have considered packages in terms of mutual adaptation tend to emphasize one side or the other.[11] While highlighting important issues, we argue that the theory and policies of computer system adoption and implementation have failed to keep pace with the challenges and dilemmas raised by the widespread diffusion of standardized software packages. We see a number of problems. First, much of the research is based on 'snapshots' that emphasize only single phases or aspects of the software package life cycle (such as implementation) and there is no attempt to follow software as it evolves, matures or crosses organizational boundaries. In this respect, we suggest focusing on the 'biography' of these technologies (see below). Second, where there has been research on package suppliers this has centred on labour process and organizational issues, such as the occupational hierarchies developing between those who design packages and those who implement them,[12] the contrasting working cultures of package and bespoke system designers,[13] and the difficulties of co-ordinating packaged software teams who are seldom co-located and distant from those who will eventually use their systems.[14] There has been little focus on how suppliers of packages manage the tension between designing system for a specific user and for a wider market. ERP suppliers have incentives to build systems that can be applied in the widest range of settings, for instance, and, therefore, design software with general or 'ideal types' of businesses in mind, even though no such form of organization actually exists. How does this process of genericification occur?

Third, there has been little focus on the relationship between supplier and user organizations, and very little is known about how package suppliers actually interact with or get to know about their users, for instance.[15] In terms of the former, it is assumed that there is little interaction between these two groups other than that brought about through initial procurement activity.[16] Even during the process of implementation when there are possibilities of interaction it is generally accepted that it is user organizations and not suppliers who are responsible for the configuration of the package and its rollout within the organization.[17] In terms of the latter, there are thought to be two different approaches for how suppliers understand user groups. One entails finding a representative organization and developing a version of the system based on this organization. The next step is to redesign or make the system more generic by identifying those 'universal' aspects of the system whilst coding out specific user features, a process that is said to be extremely difficult or even in some cases 'impossible'.[18] An alternative, and more popular approach is to base basic understandings of the user group on 'text book' models of the application area rather than on interactions with user organizations, as was typified by the development of packages like Computer Aided Production Management (CAPM).[19] This is because suppliers wish to avoid affiliating their package with any one group of organizations for fear that it will become too specialized (and therefore not marketable as a generic package).

We suggest an approach that focuses on the 'biography' of software packages. Building on work from within material culture,[20] we attempt to trace the 'accumulated history' of a software packages and show how it continues to influence the structures and practices of later adopters. In particular, the notion emphasizes the way artefacts move around (across national borders or the boundaries of several industrial sectors) and are adapted and redefined according to the needs of each new setting. The approach also highlights the various relationships and meanings an artefact that is established among one community may have for actors and communities in other settings. This perspective has obvious sympathies with Social Worlds Theory, which has discussed how distinct communities interpret and put 'boundary objects' to use.[21] It also parallels writing from Actor Network Theory, where not only humans but also artefacts and technologies are treated as actors with 'histories'.[22]

We also explore how end-users shape and exploit software through customization and other strategies and how their organizational specificities influence the evolution of the software. In this respect the notion of biography shares some characteristics with Bruno Latour's notion of chains of transformation, where changes to the technology might be seen as actors leaving behind, and the selective carrying forward, of certain aspects of the system biography.[23] It also draws on the developing 'Social Learning' perspective that has described the processes by which computer systems are integrated into existing organizational practices.[24] Two inter-linked processes in the diffusion of technical artefacts into organizations are identified within this approach. 'Domestication', which is the process whereby user organizations accommodate new technical artefacts through a process of 'learning by doing'[25] and 'innofusion', which is the process of transformation that an artefact undergoes as it diffuses into an organization.[26] These complementary dynamics emphasize the intense innovation processes involved in the 'struggle' to implement an artefact and get it to work under particular social and technical exigencies; and the more protracted process of 'design in use' where the system is transformed to match the changing organization around it. Importantly, this perspective specifically highlights the need to feed implementation experience back to future technological supply.[27]

Methodology

The first author conducted ethnographic research over a three-year period. The study was carried out at 'Big Civic', a large red-brick university in the North of England and at a large ERP Supplier. Prior to the introduction of Campus we had also followed the implementation of a number of other ERP modules at Big Civic. In line with our aim to focus on the biography of such packages we include and build on insights from this earlier work.[28] An ethnographic study suggests a long-term involvement in a particular field site, during which time a variety of methods are deployed to understand and participate in the relationships and activities ongoing in that setting. The aim of this activity is to say something about the various, often tacit, ways in which the subjects of the ethnography organize their lives. In terms of 'what' and 'who' we decided to look at we drew lessons from the sociology of science and technology and the actor network tradition. In terms of what should be studied, Latour has famously advocated that technologies should be studied not as finished artefacts (i.e. black boxes) but 'in the making', arguing that, by studying them in this way, the 'messiness' is still there for all to see.[29] By messiness he means not only those issues identified by the researcher but those that arise during the building and implementation of ICT projects. We studied, therefore, projects as they were actually being planned, built, and used. During both phases of the research we employed a wide range of qualitative methods, which included direct and participative observation of 'strategy' and technical meetings and user testing sessions.

Within Big Civic, we observed the weekly 'Sponsors Group' meetings where day-to-day technical and organizational decisions were made. The monthly 'Strategy Group' meetings were observed where issues more relevant to the future direction of the university were the focus. Project away days, comprising mainly those technical and administrative staff involved in the actual implementation of the project were also attended. Within the ERP Supplier one of Campus's testing sessions was observed, where technical teams and 'end-users' from the various pilot and early adopter sites gathered together to provide input and help shape the software. During this session, and at a subsequent 'Campus User Group' workshop, we also met and talked with programmers and analysts from the technology supplier as well as staff from the other participating universities. We were also able to conduct a number of individual and group-based semi-structured interviews, as well as more informal discussions with members of the Big Civic technical team and user community. Finally, supporting material, such as meeting notes, email exchanges and reports were also collected and analysed.

Background

Until recently, universities like many other large organizations relied on computer systems that had been developed 'in-house' and that had grown over time in an ad-hoc manner. These systems were typically maintained by dedicated university staff who, as the need arose, would develop new software or as was increasingly happening 'bolt on' commercial packages to meet changing institutional requirements.[30] While there had been little research done on the actual usability and effectiveness of these early systems, there were numerous complaints about the quality of management information that these systems provided, particularly that which was reported upwards to Government.[31] The 'MAC' initiative, which was a bespoke management information system developed during the early 1990s, was an attempt to remedy this through standardizing the way in which information was collected and reported. Widely adopted, these systems were largely acknowledged as 'unsuccessful' as many institutions quickly looked to replace them with alternative solutions.[32]

Big Civic, who were not keen to repeat their MAC experience, decided to procure a generic off-the-shelf system and thus invited a number of well-known enterprise-wide system providers to tender for the project. From the potential bidders, a large European Supplier was selected; one of the reasons for this choice was that it was the most high profile of the suppliers around, their product having established itself as the market leader. Another (rather counter intuitive) reason was that the system was known to be highly 'prescriptive' with the integration of its various 'modules' (HR, Finance, Logistics, etc.) demanding a simultaneous wide-scale business process change. This met with similar plans developing at Big Civic for organizational restructuring (see below). Finally, and perhaps more importantly, while the Supplier had little experience of working in a higher education settings it had demonstrated an intention to commit resources to re-developing its software for this new market.

The actual work of building Campus began shortly after the decision to implement the larger ERP system. The University's status in this project was as one of several 'pilot sites' or, as they were some times referred to, 'development partners'. This difference in nomenclature suggests more engagement than one might normally expect from a pilot site (the nature of this engagement and the extent to which the Supplier is able to manage the full participation of several development partners is a theme we return to in the paper). Right from the beginning, Big Civic was particularly active in the project, and encouraged by the Supplier, established a dedicated 'student management system team' ('the team') comprising experts on student administration as well computer systems. Many of this team had been (or still were) involved in the wider ERP implementation and so had some understanding of the nature of the work ahead. From Big Civic's point of view the whole project appeared to be a unique opportunity to have all the benefits of a standard software package without many of the problems (i.e. the system could shaped according to their specific needs). From the Supplier's point of view it was a good opportunity to have a British university involved, which they might use as a 'flagship reference site' to encourage other institutions to adopt their ERP solution (internal report, April 2000).

More generally, throughout Big Civic there appeared to be initially a good level of support for Campus and the deal the University had managed to broker. As part of the partnership, for instance, it was agreed that the Supplier would meet all the software costs as well as some of the implementation costs. On the project website, it was described as a 'pioneering' development between a well-respected software supplier and several prestigious universities from around the world with which Big Civic was at the forefront. The plan was eventually to merge Campus with the other ERP modules already installed, meaning that not only would Big Civic be among the first to rollout the Campus module but also it would be the first in the world to have a fully integrated 'University ERP solution'.

The Genesis of Campus

As we have said, the package investigated in this study was understood to be among the most rigid of ERP systems on the market, carrying both prescriptive and proscriptive assumptions about the nature of work and organizational process. Studies have shown us that systems with such a mature or well-defined biography do not travel particularly well.[33] Below, we describe how the new Campus module develops out of, and attempts to carry forward, this biography. The Supplier was aware of possible transferability problems and planned from the outset to commit significant resources to re-developing its software for this new market. This included tailoring existing modules to meet new requirements and, where there was seen to be a lack of functionality, the development of the new Campus module. However, before embarking on a full-scale development project, it appeared to the Supplier that there was the possibility of re-using existing software whilst building Campus. For example, the Training & Events Management module and the Real Estate module, both of which were designed around the needs of commercial organizations, appeared to offer the kind of functionality demanded by universities. However, even after much adaptation work these systems did not translate as easily into their intended setting as has been anticipated. For example, an e-mail from a Belgian University, one of the 'pilot sites' involved with Big Civic in the development of the Campus module, describes some of the limitations of re-using the existing software:

> Until now the Real Estate module has always been referred to for student housing. This only contains the functionality to 'let' rooms (very commercial). For some universities this is not enough. Student rooms are often part of student aid. A lot of extra activities have to be organized in association with this (e.g. meals).

There were also complaints from many across Big Civic:

> . . . the Real Estate Module is so far removed from our requirements that [the Supplier] would do better to start again from scratch than try to adjust the existing module. The most obvious shortcomings of 'Real Estate' for us is that the module is designed for the commercial sector where long term lets of 12 months or more are standard. It is not designed for the levels of volume and turnover that characterise the student market and, even more obviously, the conference market. In short, Real Estate does not set out to be a retail booking system, which is what we are looking for (internal memo from the Estates Department).

One problem concerned the incommensurabilities between the roles and responsibilities of actors embodied in the system and those within Big Civic. ERP systems are structured around general notions such as 'supplier', 'customer' and 'employee' and while these may share some of the characteristics of actors found in universities, they do not map straightforwardly. None of these categories appeared to fit with the notion of a student, for example. In the Real Estate module, therefore, which was being adapted for the management of accommodation on campus, the student had to be in effect conceived of as a ‘special type of employee’, one who was undertaking a long-term training course and thus permanently renting a room. Unsuprisingly, this raised tensions among those with responsibilities towards student administration who found that they could not account for many of the circumstances and characteristics of the students they routinely dealt with. Indeed, some months later the Supplier would admit to Big Civic and the other pilots how re-using software was not appropriate in the case of universities, nor would it be possible to simply move the universities towards the organizational assumptions embodied in the software. One member of the Big Civic team describes the issue rather elegantly:

> Part of our problem in the beginning, with them and with other consultants, [was that] we didn’t want to go through the ‘sausage machine’ and they kept trying to push us through and it took a long time for them to realise that we wouldn’t go . . . . (interview, 09/2000)

Localization through Standardization

The biography approach draws attention to how these technologies are always in a constant process of transformation and translation. Suppliers will attempt to reconcile the gulf between their solutions and the specific contexts of adopters in a number of interesting ways. One method is to localize the system (not to the many idiosyncratic practices of the adopters) but to ‘common processes’ throughout the university. Below, we discuss how at Big Civic and the other pilots no such processes existed and thus had to be created. Under pressure the Supplier set about a major software writing effort in order to move the Campus module from the typical or ‘generic’ ERP user to a more specific university user. The actual work of adapting and expanding the software to the new context of the university was organized as a complex chain linking Big Civic, the other pilot implementation sites, with analysts and programmers (who were adapting the software). Individual requests for changes to the system (for example, the addition of new functionality) were passed along the chain. In this adaptation work, information about existing management and administration processes would, theoretically at least, be collected and passed back down the chain to the programmers, who would pass back code based on their understanding of the information they had received. In practice, however, the analysts and programmers often received incomplete or ambiguous responses to the questions they were asking. One issue that the Supplier continually came up against was that ‘there was no university there to fit to the system’.[34]

By this we mean that for some functions of the university there are no clearly articulated institution-wide policies or standards, with individual departments often working according to their own informal practices—a situation that is typical of what Ian McNay describes as the classic ‘collegial academy’.[35] For example, in one meeting conducted on the Supplier’s premises and attended by Big Civic and a number of other pilots, there was a session where institutions were asked to clarify and explain their policies for student registration. What emerged was that there was a diversity of processes for carrying the same task within each institution: ‘We have three different ways of handling credit points’ (Large Campus University). Moreover, some activities were clouded as to who did what and when: ‘What I can determine is that you don’t have a clear cut process, it is informal’ (Supplier Analyst summarising one particular conversation). Others activities appeared to be so complicated as to be shrouded in mystery—when asked to explain how their university assigned students to particular faculties and departments, for instance, a Large Campus University representative somewhat ironically described how: 'It is magic. I don't know how to clearly explain our situation'.

Within Big Civic there were similar problems. We observed, for instance, how the implementation team attempted to uncover the recent history of practices regarding the handling of student fees so that they might be explained to the Supplier and included in the module. In one conversation, the team is trying to work out the process for setting and administering 'fee categories', which were categories that had been established on the existing MAC system. It was not obvious, however, why the categories were there or who had decided which course was applied to which category:

> Every degree programme is currently assigned to a fee category within MAC. However, the process by which new programmes are allocated to categories is unclear and haphazard. The fee category is an important parameter in the new system . . . . (Internal document)

Further investigations provided no clues and, thus, the team concluded that the staff responsible for inputting fee categories was just simply 'making them up'.

As a result, the actual means by which the module was localized for use in Big Civic represented something of a 'bootstrapping problem'.[36] Fitting the Campus module to the University meant first drawing upon institutional-wide rules and procedures. These as we have described were not always available. Designing new sets of common institutional standards (such as a fees policy) that Campus might draw upon was further complicated, moreover. The team did not as yet have the module to guide or direct them in drawing up new policies and standards and thus found it difficult to set out and plan institution-wide processes while the system was not available. At the same time, the Supplier was unable to fully localize the system until this process of standardization had been carried out. In many cases both parties had to 'imagine' how the other might work.[37]

Building a Global Product

Reconciling this gulf between the generic model and the specific mode of use required by Big Civic was not the only barrier to the implementation of the module. Other problems originate from the Supplier's intentions to market the module around the world as a potential 'global product'. Consider the following extract:

> Campus is intended to be a global product. In order to facilitate this, the software is being designed to be as flexible as possible to allow individual universities to configure the functionality according to their own processes. Because the basic design has to accommodate university structures and processes across many countries, much of the functionality can be configured in more than one way and the overall functionality is likely to be much wider than that required for any one country. (Big Civic internal report)

The module therefore was to include the 'local standards' of all of the participating sites and where specific features could not be included each site would carry out their own individual customization work. In other words, the Supplier was not only attempting to move the module from a generic user to a specific user, but it was attempting to move it back again to a more general form of 'university user'. Interestingly, one of the methods used by the Supplier to do this was to invite the pilots and other 'early adopters' to regular testing sessions of the module. One of the authors observed one such meeting, where the aim was to find where there might be some common requirements among universities, and if not how these might be constructed.

Constructing General University Users

One method of arriving at common needs was through the development of what the Supplier called 'generalizable concepts'. These appeared to be concepts that were sufficiently flexible to be able to work across multiple sites but were also not so weakly structured as to lose their shared identity. Below, we discuss an attempt to construct one such concept for the practice of 'holding'. Holding is the process by which a student might be blocked on the system from re-registering for a new academic term because of outstanding debts to the university (tuition fees, unpaid rent, library fines, etc.) or through failing an exam or coursework. The issue being discussed below is whether holds should be input manually by administration staff or whether they should be automatically triggered by Campus. According to the Supplier manual inputs are a 'system limitation' and if the pilots could only arrive at a common set of understandings about the holding process, then the procedure could be automated. In the meeting, one analyst asks for comments on what currently happens at each institution:

> *Supplier Analyst*: Students with bad marks. What do you do with them, leave them in limbo or give them a second chance?
> *Southern University*: Depends on timing, if just before a session and there is no chance of them bettering their mark, then we refuse them. Or, alternatively, we could say we've not decided yet. That is not a hold but a 'waiting status'.
> *Technology University*: If you're doing something that might pick up your grades?
> *Supplier Analyst*: I wouldn't call that a hold, that's a 'provisional situation'.
> *Rural University*: We have a 'partial hold', so holds affects some things. . . .
> *Large Campus University*: Isn't that a 'half-hold'. . . .

The discussion goes on for some time and finally everyone (including the analyst) appears to be in a state of confusion as to what a hold might be (a 'waiting status', a 'provisional situation', 'a half hold . . .').[38] Reaching common understandings and developing generalizable concepts is far from straightforward. Interestingly, it is the Supplier and not the universities who were keen to have these discussions. For them, such a process is useful as it allows the construction of a robust concept and thus one that is applicable to the widest variety of higher education institutions. In contrast, the participants at the testing session were becoming increasingly frustrated by the attempts to understand each and every difference among the universities present.

Incorporating Specific Contexts

In these initial stages of Campus's biography the module was all-inclusive as the aim was to build a generic package for use across many types of higher education institutions and not around the specific needs of any one adopter. However, some time later, it appeared that in order to produce a working solution that there was a need to build a specific and local context into the software.[39] It was widely thought, for instance, among those participating in the pilot project that Big Civic and 'Large Campus', an American University also participating as a pilot site, had been particularly successful in ensuring that many of their features were included in the software. This was apparently because they were to be the first to 'go live' with the module. A Big Civic internal report describes how:

> Although [the Supplier] is aiming to build Campus with a 'global common core' to the software, [Big Civic] has been able to influence the content of that global core product to its benefits, along with the UK specific requirements. This should ensure

> that the amount of 'customer specific' development and configuration is kept to an absolute minimum.

However, towards the latter stages of the project, Big Civic found that though it had been able to shape the content of the module to its benefit that it too was beginning to lose its influence in light of pressure from Large Campus. The Supplier it seemed had problems coping with wide-ranging and sometimes diverging needs of the pilot sites:

> [The Supplier] has problems: too many requests from the pilots, together with [Large Campus's] demands for changes, means that they have more demands than resources and are trying to prune down on what they need to give to us go live with. (Big Civic Project Director speaking at an Away-day)

Part of the problem as the Project Director saw it was that the Supplier did not yet have a clear vision of a general university user: 'There are lots of people from around the world asking for different things and because *they don't have a model of university to work with*, they can't decide on what is important' (our emphasis). Moreover, the Supplier was finding it increasingly difficult to continue to resource the production of generalizable concepts, particularly when many appeared to work across a few sites only. Therefore, the universities were drawn into a struggle with the Supplier and with one another over the inclusion of their specific needs. This as described by the Project Director at Big Civic led to a 'push and shove' between them and Large Campus.

Campus Takes Shape

The nature of the struggle between Big Civic and Large Campus University was over the scope of Campus. The American University was pushing the Supplier to include functionality to cope with differing forms of student enrolment and registration. They argued that the inclusion of advanced features such as 'self-service' functionality and links to Customer Relationship Management (CRM) software would suit their more 'market oriented' admission process. The Project Director at Big Civic, however, saw this as having implications for their student management processes:

> Rather than a UCAS [University Centralised Admissions System] type system, [Large Campus] has a competitive marketplace and therefore wants online student registration. [The Supplier] has finite support and [Large Campus's] needs is more complicated than ours . . . [Large Campus] also plonked on the table more requests for CRM and that has blown the whole thing apart. If [the Supplier] meets this request, then that is less for us. (Notes from an away-day meeting)

His concerns were that the module was about to take a direction that would not be workable in the context of Big Civic. Although initially open to the possibility, and team members from Big Civic had been among those asking for the inclusion of such functionality, colleagues in the Central Administration and throughout the wider University were less convinced that self-service registration or CRM was desirable in their context.[40] On top of this, much of the new technology (like CRM) appeared to be at odds with their more 'handicraft' and 'paper-based' approach to the registration and management of students on campus. As both sets of needs could not be reconciled within Campus the supplier had to prioritize one set of requirements over another; choosing as some internal critics have described to build the system according to the needs of the larger US market.

Accommodating Unwanted Functionality

It has been suggested that computer systems that become so 'thoroughly imbued' with local idiosyncrasies can only work at one site.[41] While suppliers of packages are careful to ensure that a package does not fully mirror the needs of one particular adopter, there are times when adopters have to accommodate the specific requirement of an earlier adopter into their own organizations. These accommodations provide for interesting tensions and are worth exploring. In order to implement Campus, Big Civic had to live with features that were at odds with its own requirements. One example, and there were many others of this kind, was the part of the system used to record applications by prospective students. This was built according to procedures common within American universities, and as applicants there are required to submit 'application fees', the system automatically generates an 'accounting record' for each new prospective student so that the appropriate information can be logged. There is no similar fee requirement in Britain therefore leaving Big Civic with the problem of deciding what to do with all the unwanted accounting records, some 30,000 being generated each academic year.

Indeed, the presence of so much unwanted functionality presented something of a paradox for Big Civic. On one hand, it was important that inappropriate aspects of the system were modified to ease concerns of prospective users who were already questioning the suitability of Campus and departmental managers who wanted evidence that the new system was better than the existing one. On the other, there were also compelling reasons why Big Civic should simply accept and accommodate inappropriate aspects rather than rewrite the code each time. If the module was to be localized too much then there was the risk that the University would be unable to make use of later upgrades and new functionality released to the worldwide Campus user community (one of the primary reasons why an ERP system was acquired in the first place). In the case of the accounting records, therefore, these were simply stored on the system as they were generated—a fix that ultimately reduces the efficiency of the system and goals of reducing data redundancy. Within Big Civic, deciding whether to localize the module or maintain it as a global product was always a point of tension—a problem that has become known as the 'package paradox'.[42]

Conclusions

Organizations are increasingly reliant on the use of packaged software solutions because of the economic benefits of commodified solutions, the interoperatibility benefits of standard platforms and the desire to align with best practice. However, there is often a large gulf between standardized generic solutions and the specific contexts, practices and requirements of particular user organizations. While there is a growing focus on the incommensurabilities between technology and organization, much of this research is limited (focusing only on single aspects of the package lifecycle) and simplistic (emphasizing either the 'global' or 'local' aspects of the package). Our argument has been that we need better mechanisms for understanding the coupling between technical and institutional change. This is in terms of the implications of software evolution for organizational change; by this, we mean the various characteristics a package accumulates through previous stages of design and the process by which its history or 'biography' continues to shape later adopting organizations. We also need to understand the influence of existing organizational practices on software design; here we intend the factors that mediate an organization's ability to shape and domesticate software and how their demands can come to influence the evolution of the package itself.

In the article we have highlighted a number of stages that occurred in the biography of this ERP package. In the first, we described how incommensurabilities between the package and the setting were highlighted. Indeed, reconciling the package with the context of universities provided for an acute problem. There are fuzzy boundaries between universities and organizations more generally; while they engage in many similar activities they are arguably 'different'.[43] It is difficult, for instance, to translate software to a new setting when packages embody conflicting categories and institutions have good reasons for continuing to use their own rather than accepting the generic classifications embodied in the software.[44] This makes the use of standard software all the more difficult. The depiction of students as 'special types of employees' and the subsequent tensions this provoked suggests that this is far from a trivial problem and not one that will go away. Standard software is unable to support the diversity of classifications at use in a particular site and therefore some form of standardization or convergence with the package is inevitable.

The second stage we described was the actual means by which the module was translated for use in Big Civic and how both the software and the institution had to be created, but each depended on the other and this represented something of a 'bootstrapping problem'. Indeed, this was a familiar theme among many of the other institutions involved in the project. It is one of the reasons why the Supplier ran into difficulties when attempting to move the module from a generic to a specific university user ('they don't have a model of the university to work with'). Similarly, in the third stage, when attempting to fit the module to the demands of all the pilots, they found themselves flooded by requests ('lots of people from around the world asking for different things . . . they can't decide on what is important'). Here the Supplier is forced to make a choice between the advantages of increasing a package's scope and specificity (its greater utility and fit to the user organizations) and the cost advantages of increasing its market size. Increasing the range of functions within Campus gives benefits and potential value to the users. However, embodying evermore local knowledge and presumptions about organizational practices within the software will have implications for its applicability and fit in other organizational settings. Therefore, they begin a fourth stage, the process of developing the system for a generic university user (i.e. the production of generalizable concepts). This stage was particularly interesting as it involved a process of genericification as the supplier attempted to replace idiosyncratic features with 'common' ones, what Schumm and Kocyba refer to as a process of 'decontextualization' and 'recontextualization'.[45] While this had initially worked well, this genericification process was beginning to be too resource consuming and we saw the fifth and final stage, where the module moved towards one particular design (that of Large Campus). This led to strains on the relationships between the supplier and Big Civic, who had agreed to pilot new software predicated on the belief that they could influence the shaping of the package in some way through allowing the software to be designed around their particular institution. Once the Suppliers attempted to make the module more generic, however, they experienced a 'loss of control' as their specific features were 'designed out' out of the system.

In summary, what we have presented here is a first reflection on the biography of software packages and on the various tensions and complexities that surround the design, use and evolution of software packages. More comparative studies are needed of: (1) the strategies and decision making processes of those adopting software packages and how they assess and make sense of the wide range of alternatives and options available, both before and during implementation (and in later decisions about whether to procure further modules); (2) how suppliers make strategies and take decisions about product design and markets and how these influence the uptake and eventual fit of a package;

and (3) how suppliers manage the tension between designing system for a specific user and for a wider market.

Acknowledgement

The authors would like to thank all those who provided comments and feedback from the earlier versions presented at the ACM Symposium on Applied Computing in Madrid (March 2002) and EASST Conference, York, UK (August 2002). The research presented here was funded under the UK's Economic and Social Research Council (Award No. R000223276). We also thank those at Big Civic University and the Technology Supplier for allowing this study to take place. Neil Pollock would also like to thank John Goddard, James Cornford and Luciana D'Adderio for their intellectual and practical support whilst he was conducting the 'students as users' project. We would finally like to thank the anonymous referees for their comments and forcing us to clarify our arguments.

Notes and References

1. T. Davenport, *Mission Critical: Realising the Promise of Enterprise Systems* (Boston, Harvard Business School Press, 2000).
2. P. Beynon-Davies, *Information Systems* (Basingstoke, Palgrave, 2002).
3. In certain sectors major companies may possess and interact electronically with one particular kind of system, meaning that an organization supplying goods to these companies may have no option but to implement the same system, A. Parr & G. Shanks, 'A Model of ERP Project Implementation', *Journal of Information Technology*, *15*, 2000, pp. 289–303.
4. T. Davenport, 'Putting the Enterprise into the Enterprise System', *Harvard Business Review*, *76*, 4, 1998, pp. 121–132.
5. O. Hanseth, C. Ciborra & K. Braa, 'The Control Devolution: ERP and the Side-effects of Globalisation', *The Data Base for Advances in Information Systems*, *32*, 4, 2001, pp. 34–46.
6. Davenport, *op. cit.*, Ref. 1, p. 152.
7. B. Light, 'The Maintenance Implications of the Customisation of ERP Software', *Journal of Software Maintenance and Evolution: Research and Practice*, *13*, 2001, pp. 415–429.
8. E. Wagner & S. Scott, 'Unfolding New Times: The Implementation of Enterprise Resource Planning into an Academic Administration', Working Paper Series, No. 98. Department of Information Systems, London School of Economics, UK, 2001.
9. N. Pollock, 'The Virtual University as Timely and Accurate Information', *Information, Communication & Society*, *3*, 3, 2000, pp. 349–365.
10. The term 'mutual adaptation' stems from W. Orlikowski, 'The Duality of Technology: Rethinking the Concept of Technology in Organizations', *Organization Science*, *3*, 3, 1992, pp. 398–427. In terms of ERP, it is developed in the work of O. Hanseth & K. Braa, 'Technology as Traitor: Emergent SAP Infrastructure in a Global Organisation', in: C. Ciborra (Ed), *From Control to Drift: The Dynamics of Corporate Information Infrastructures* (Oxford, Oxford University Press, 2000).
11. Hanseth & Braa, *op. cit.*, Ref. 10, p. 143. However, see a later paper by the same authors where they attempt to reconcile this tension between the application of universal technologies and their adaptation at local sites, O. Hanseth & K. Braa, 'Hunting for the Treasure at the End of the Rainbow: Standardising Corporate IT Infrastructure', *Computer Supported Cooperative Work*, *10*, 2001, pp. 261–292.
12. A. Friedman, *Computer Systems Development: History, Organisation and Implementation* (Chichester, Wiley, 1989).
13. S. Sawyer, 'Packaged Software: Implications of the Differences from Custom Approaches to Software Development', *European Journal of Information Systems*, *9*, 2000, pp. 47–58.
14. E. Carmel & R. Agarwal, 'Tactical Approaches for Alleviating Distance in Global Software Development', *IEEE Software*, March/April, 2001, pp. 22–29.

15. An exception is the article by Scott and Kaindl who focus on how suppliers and users work together to develop new ERP modules. J. Scott & L. Kaindl, 'Enhancing Functionality in an Enterprise Software Package', *Information & Management*, *37*, 2000, pp. 111–122.
16. J. Bansler & E. Havn, 'Industrialised Information Systems Development', CTI Working Paper No. 22, Technical University of Denmark, 1996.
17. *Ibid.*
18. *Ibid.*, p. 6.
19. J. Webster & R. Williams, 'Mismatch and Tension: Standard Packages and Non-standard Users', in: P. Quintas (Ed.), *Social Dimensions of Systems Engineering* (Hemel Hempstead, Ellis Horwood, 1993).
20. See A. Appadurai (Ed.), *The Social Life of Things: Commodities in Cultural Perspective* (Cambridge, Cambridge University Press, 1992). The notion that software has a 'biography' builds on earlier work by Robin Williams, see T. Brady, M. Tierney & R. Williams, 'The Commodification of Industry Applications Software', *Industrial and Corporate Change 1*, 3, 1992, pp. 489–514; R. Williams, 'Universal Solutions or Local Contingencies? Tensions and Contradictions in the Mutual Shaping of Technology and Work Organization', in: I. McLoughlin & M. Harris (Eds), *Innovation, Organizational Change and Technology* (London, ITB Press, 1997), pp. 170–185. Kathryn Henderson also discusses a similar notion in her description of the 'career' of a prototype; K. Henderson, *On line and On Paper: Visual Representations, Visual Culture, and Computer Graphics in Design Engineering* (Cambridge, MA, MIT Press, 1999).
21. S. Star & J. Griesemer, 'Institutional Ecology, "Translations" and Boundary Objects: Amateurs & Professionals in Berkeley's Museum of Vertebrate Zoology, 1907–39', *Social Studies of Science*, *19*, 1989, pp. 387–420.
22. B. Latour, *Pandora's Hope* (Cambridge, MA, Harvard University Press, 1999).
23. *Ibid.*
24. R. Williams, J. Stewart & R. Slack (Eds.), *The Social Shaping of Multimedia, COST A4* (European Commission DGXIII, Luxembourg, 1999).
25. K. Sorensen, 'Learning Technology, Constructing Culture: Socio-technical Change as Social Learning', STS Working Paper No. 18/96, Centre for Technology & Society, Trondheim, 1996.
26. J. Fleck, 'Configurations: Crystallizing Contingency', *Human Factors in Manufacturing*, *3*, 1, 1993, pp. 15–36.
27. *Ibid.*
28. J. Cornford & N. Pollock, *Putting the University Online: Information, Technology and Organisational Change* (Milton Keynes, Open University Press, 2003).
29. B. Latour, *Science in Action: How to Follow Scientists and Engineers Through Society* (Cambridge, MA: Harvard University Press, 1987).
30. G. Bull, C. Dallinga-Hunter, Y. Epelboin, E. Frackman & D. Jennings, *Information Technology: Issues for Higher Education Management* (London, Jessica Kingsley, 1994).
31. A. Goddard & P. Gayward, 'MAC and the Oracle Family: Achievements and Lessons Learnt', *Axix*, *1*, 1, 1994, pp. 45–50.
32. Pollock, *op. cit.*, Ref. 9; J. McLaughlin, P. Rosen, D. Skinner & A. Webster, *Valuing Technology: Organisations, Culture and Change* (London, Routledge, 2000).
33. M. Berg, *Rationalizing Medical Work: Decision-Support Techniques and Medical Practices* (Cambridge, MA, MIT Press, 1997).
34. This point is drawn from a discussion with James Cornford.
35. McNay describes UK universities, for example, as characterized by a 'lack of co-ordination', the 'absence of regulations' and the 'invisibility' of much that happens. I. McNay, 'From the Collegial Academy to Corporate Enterprise: The Changing Cultures of Universities', in: T. Schuller (Ed.), *The Changing University?* (Buckingham, Open University Press), pp. 105–115. On this point see also Cornford & Pollock, *op. cit.*, Ref. 28.
36. G. Bowker & S. Star, *Sorting Things Out: Classification and its Consequences* (Cambridge, MA, MIT Press, 1999).
37. There were many questionnaires, for example, sent to the University during the initial requirements gathering phase which began with the statement 'Imagine the case where…'.
38. Such is the confusion that laughter breaks out and one participant shouts how all of this is beginning to '… sound like a philosophy class!'.

39. Berg, op. cit., Ref. 33.
40. For more information on this see N. Pollock, 'The "Self-Service" Student: Building Enterprise-Wide Systems into Universities', *Prometheus*, *21*, 1, 2003, pp. 101–119.
41. Berg, op. cit., Ref. 33, p. 105.
42. M. Tierney & R. Williams, 'Issues in the Black-boxing of Technologies', Edinburgh PICT Working Paper No. 22, Edinburgh University, Edinburgh, 1991.
43. Cornford & Pollock, *op. cit.*, Ref. 28.
44. The unique identity of the student and the universities' refusal to conflate this with the notion of 'employee' is a good example of a conflicting category. See for instance G. Bowker, 'Biodiversity Datadiversity', *Social Studies of Science*, *30*, 5, 2000, pp. 43–83.
45. W. Schumm & H. Kocyba, 'Re-Contextualisation and Opportunities for Participation—The Social Shaping of Implementation', *Proceedings of the COST A4 Workshop*, Gilleleje, Denmark, April 1994.