

The 'Biography' and Evolution of Standardised Software Packages

Project Outline

After more than 30 years of software development for an ever-growing variety of institutional and organisational settings, few large-scale information systems are developed completely from scratch. Rather, most software applications are constructed by adapting existing 'packages' to new organisational contexts and settings. Generic software packages, such as Enterprise Resource Planning (ERP) systems, cover the fullest range of organisational activities and processes and are adopted with the aim of achieving substantial cost savings as well as improved access to 'tried and tested' solutions, new releases, and an opportunity to update procedures and align them with perceived 'best practice'. However, while organisations choose packages because of their economic benefits they are potentially a costly and high-risk strategy. Systems seldom translate easily across boundaries, whether this is between organisations within the same sector, between industrial sectors or between public and private sector organisational forms (cf. Walsham, 2001, Ciborra et al, 2000). There is often a gulf between the system and the specific contexts, practices and requirements of particular user organisations. Amongst the many issues generic packages raise, of particular concern to practitioners is the choice between conducting expensive 'customisation' work on standard solutions or undergoing unwanted organisational change in adapting their practices to models of work and organisational process embedded in the software. The development of COTS software differs from conventional bespoke information systems in that packages are designed for a market and not for a specific customer. Suppliers have incentives to build systems that can be applied in the widest range of settings and design software with general or 'ideal types' of businesses in mind (even though no such form of organisation actually exists). As concerns rise concerning the incommensurability of systems and contexts there are demands for solutions which are already partially adapted to particular business settings (i.e., 'semi-generic' packages, Webster & Williams 1993) and for user-involvement in the shaping of packages. Alongside the adoption of these systems, then, there is an equally important story of innovation within supplier organisations and collaboration with package adopters.

Why Software Packages?

Large COTS software packages (such as ERP, customer relationship management (CRM) and other types of integrated financial and administrative information systems) represent a substantial part of organisational IT expenditure. Over 60% of organisations rely upon COTS software packages (Parr & Shank, 2000). Yet despite their wide diffusion these packages are still only partially understood - much of the research has taken the perspective of the adopting organisation. We know very little about how competing pressures for standardisation and differentiation are resolved both in user and supplier organisations. An important question thus concerns how the economic and dependability benefits of packages can be combined with strategies to improve the fit between standard solutions and specific user organisations and classes of similar organisations. The proposed work builds on a long-term research programme at the Universities of Edinburgh and Newcastle on the social shaping of computer systems (cf. Brady et al, 1992; Williams 1997; Pollock 2000; D'Adderio 2001, in press; Hartwood et al. 2002; Cornford & Pollock, 2002).

Research Issues

The implementation and Use of Packages

The wide diffusion of COTS software packages owes much to the supplier strategy of taking a technology that has worked in one place and adapting it to work elsewhere, and, in principle, 'everywhere' (cf. Brady et al, 1992). This is evidenced by the fact that ERP packages were conceived for and used by manufacturing firms before being applied within non-manufacturing and, more recently, non-commercial settings and contexts (health care, public sector and higher education). ERP systems have become so successful that they are now commonly described as the de facto standard for the replacement of legacy systems and it is said that some companies find it impossible to work without one (Parr & Shank, 2000). Moreover, in order for suppliers to reap the economies of scale these systems must function in new settings in much the same way as they have functioned in all other settings (Koch, 2001, Walsham, 2001). Several studies have considered the 'impact' such standard solutions have on organisations. Davenport (1998), for instance, discusses how ERP systems typically force adopters to replace informal way of working with the more formal 'business process templates' embodied in the software. The benefits of this kind of standardisation are that suppliers have to cater for only limited amounts of variation in product maintenance and new upgrades; for user organisations the benefits are having simple and

guaranteed upgrade pathways. The downsides for suppliers are foregoing the high value-added markets for customised solutions; for user organisations 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' 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'. One study, for instance, estimates only 5% of organisations actually attempt significant customisation (Davis, 1998). That is, rather than attempt to reconfigure each and every aspect, implementation teams simply accept those 'default' features already embodied within systems, what one author has called the 'power of default' (Koch, 1999).

There is, however, no consensus about just how much customisation can be carried out. Davenport 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' (2000, p152). Light similarly points out that some organisations cannot completely adopt the standard model and therefore have no choice but to attempt customisation (2001). Other, more ethnographic based research has argued that even the most prescriptive of systems are typically 'localised' by adopters and end-users (D'Adderio, 2001). Scott & Wagner (in press) 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'. In summary, then, one body of literature emphasises how adopters end up fitting their organisation to the system (rather than the other way around) and another pays particular attention to the 'workarounds' (Pollock, 2000) and other strategies that users deploy to adapt technologies to the local setting. A third strand has sought to reconcile these two positions through emphasising how technology and organisation are often brought into alignment through a combination of quite complicated organisational change and software configuration, a process which is sometimes known as 'mutual adaptation' (Orlikowski, 1992; Hanseth & Braa, 2000; McLaughlin et al, 2000).

While highlighting important issues, we would argue that the focus of much of this research has failed to fully keep pace with the challenges and dilemmas raised by standardised software packages. Much of the current research, limited by prevalent short-term funding models, is based on 'snapshots' that emphasise only single phases or aspects of the package lifecycle (such as implementation). There is no attempt to follow what we describe here as the 'biography' of software as it evolves, matures and crosses organisational boundaries. Therefore, we know very little about innovation within software suppliers and the methods by which packages are generated or adapted to new sectors or settings. There have been few studies of how supplier product development strategies influence the uptake and eventual fit of a package (cf. Quintas, 1994). Nor has there been much attempt to understand how user organisations feature in the evolution of packages and whether (or how) local customisations, for example, are fed back into the standard package (cf. Light, 2001) and the questions as to why they procure certain types of packages. In this research we are interested in understanding the different strategies and decision-making processes of those adopting software packages and also with how they attempt to shape these systems. What we mean here is the process by which users assess and make sense of the wide range of alternatives and options available. Whether to procure one of the more 'rigid' of systems on offer or a more flexible ERP alternative (cf. Koch, 1997). What level of customisation to carry out. The choice about whether to fit the system to the organisation or the supplier standard, a dilemma that has been described as the 'package paradox' (Webster & Williams, 1993). On one hand, it is acknowledged that organisations find it difficult to critically assess and evaluate the range of packages on offer. There is, as we have suggested, a growing awareness of the costs of ending up with the 'wrong' solution and this is provoking uncertainty among user organisations. Yet, on the other, this appears to have done little to deter the uptake of packages. In terms of the shaping of these systems we are interested in how standard solutions are reconciled with the specific contexts, practices and requirements of user organisations and how the organisation's specificities, in turn, influence the software evolution.

The Design and Evolution of Packages

Questions concerning the evolution of software have largely been ignored. Discussions of suppliers have in the main focused on labour process and other organisational issues surrounding programmers (Friedman, 1989, Sawyer, 2000). Very little is known about how package suppliers

actually get to know about their users (Kjaer & Madsen, 1997), and how such interactions actually influence package design. In much of the research it is assumed there is little communication other than that brought about through initial procurement activity. In terms of the design of packages, for instance, Bansler & Havn (1996) argue that basic understandings tend to be based on 'text book' models of the application area rather than interactions with user organisations. This is presumably because suppliers wish to avoid affiliating their package with any one group of organisations for fear that it will become too specialised - and therefore not marketable as a generic package. While not discounting these findings, we shall investigate the hypothesis drawn from recent ESRC sponsored research on the development of SAP's Campus Management system (Pollock 2003) that both supplier claims and user demands are generating new pressures for 'semi-generic' solutions. This in turn is generating a number of interesting issues and tensions. One aspect of these differentiation strategies has been the setting up of 'user groups' (cf. Scott & Kaindl, 2000), which are networks of user organisations who meet regularly with suppliers. As the core issue in this research, then, we shall focus on the supplier-user nexus as revealed, for instance, by the operation of these groups. We will investigate their strengths, and weaknesses, and see what lessons can be learnt for the development and use of packaged software more generally. Research questions here concern:

Firstly, we wish to understand how different suppliers manage the tension between designing systems for a specific user and for a wider market. This is important whether a package is being redesigned from a generic to a niche specific solution or whether it is being translated from one sector to another, or 'upgraded' from custom built software to a generic system. This latter strategy is particularly interesting as it involves some form of 'generic-ification' as suppliers attempt to replace idiosyncratic features with 'common' ones, what Schumm & Kocyba (1997) have referred to as a process of 'decontextualisation' and 'recontextualisation'. Unsurprisingly, this provides strains on the relationships between suppliers and user organisations, as many users will have agreed to act as 'pilot sites' for new software predicated on the belief that they can influence the shaping of the package (through allowing the software to be designed around their organisation). Typically, once suppliers attempt to make their products more generic user organisations experience a loss of control as their specific features are 'designed out' of the system (Pollock et al, in print, Ellingsen, 2002). Secondly, we will investigate how suppliers make strategies and take decisions about product design and markets and how these influence the uptake and eventual fit of a package. These are choices between the advantages of increasing a package's scope and specificity (its greater utility and fit to the user organisation) and the cost advantages of increasing its market size. Increasing the range of functions within a package, for instance, gives benefits and potential value to the user. However, embodying evermore knowledge and presumptions about organisational practices within the software will have implications for its applicability and fit in other organisational settings. The simple and 'discrete' application can be readily applied in many settings whereas the most complex integrated applications will 'have a market of one' (Williams, 1997).

Research Approach

How will we address the research issues that concern us? While building on earlier studies that have examined the mutual adaptation of technology and organisation we will attempt to 'close the loop' by investigating the 'biography' of software systems. Drawing on work from material culture (cf. Appadurai, 1992), we suggest an approach that follows the actual packages themselves as they evolve and mature, progress along their lifecycle, and move across sectoral and organisational boundaries. In particular, we will explore the 'accumulated history' of generic or sector specific solutions and how their history is reconciled with the contexts, practices and requirements of a niche specific or new sector specific user organisations and with how, in turn, the specificities found here influence the package evolution. The research also builds on the developing 'Social Learning' perspective (Williams, Stewart & Slack, 1999), which has described the processes by which computer systems are integrated into existing organisational practices. Two interlinked processes are identified in this perspective: **domestication** - the process whereby user organisations accommodate new technical artefacts through a process of 'learning by doing' (Sorensen, 1996); and **innofusion** - the process of transformation that an artefact undergoes as it diffuses into an organisation (Fleck, 1993). These complementary dynamics emphasise 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 organisation around it. This perspective

specifically highlights the need to feed implementation experience back to future technological supply (ibid.).

Methodology

The approach that we have adopted is a comparative one, which analyses the biography of a number of software packages as they move longitudinally across similar organisations, from one national context to another, and from the private to public sector. Analysing the evolution of software along its life cycle and across sectors and countries will provide insights into the dynamics of domestication and innovation in a number of organisational and sectoral contexts. The selected packages are at different stages in their biography and are characterised by different levels of product maturity and standardisation. Each study raises different issues:

the **first** concerns the post-implementation development of an ERP system within a large hi-tech, manufacturing environment. The challenges faced here include lobbying the supplier to further enhance their system to take account of a specific legislative and operative environment as well as different levels of product complexity;

the **second** concerns the further development of a software package for the administration of university campus accommodation. Developed around the needs of a Scottish university and later adapted to fit other UK sites, the supplier now intends to offer the package to institutions abroad. The issue here is how the supplier reconciles the costs of increasing the scope of the system with the cost advantages of increasing its market size;

the **third** concerns the future development of an ERP system and its adoption within Universities. The issue here is how one University in a joint venture with the supplier is able to shape a mature and stable system for this new setting;

the **fourth study** is a CRM system developed for the banking sector and now being adapted for use in a local authority). The issue here is the extent to which a system developed in a commercial setting translates for use in a public sector context.

Each study will investigate the socio-technical history of each package, the implementation of the package in the user setting, the further development of the package in the supplier organisation, and the interactions between the user and supplier organisations. Consequently, the research programme will have four elements:

1) Survey of Software Package Development

We wish to understand the history of our chosen packages and the various interactions that played a role in their shaping. In some cases we will build upon earlier work to develop a longitudinal insight and where there are gaps in our knowledge we will analyse historical data. This will be made up of searches of technical publications, such as company reports, which will provide a rich source of technological detail. Wherever possible, and guided by industrial interviewees and our own previous research in these settings, we will use company archives to access past documents.

2) Interviews with User Organisations

In this part of the research we are concerned with the use and further innovation of software packages in specific organisational and sectoral settings. We intend to undertake 12 to 15 in-depth semi-structured interviews in each user organisation, where we will talk to both technical and managerial teams. Where possible we will carry out participant observation of particular phases of the projects. Interviews will be taped and transcribed.

3) Interviews with Supplier Organisations

We will conduct interviews with personnel in supplier organisations and other actors involved (e.g. implementation consultants) in the current evolution of the selected software packages. This will involve approximately 20 interviews which will seek to identify the policies and strategies with respect to the current and future development of each package. Initial contacts have already been identified through our previous research and more will be added through 'snowballing' as the research progresses.

4) Study of User-Supplier Interaction

Here we have to consider how to study interactions between user and supply organisations. We have decided that the best means of access is through studying informal user communities and the more formal 'user groups'. To work through such communities as our previous research has shown us will allow us to study the user organisations in terms of the differences between them as well as the strategies developed by Suppliers to 'manage' these differences. Analysing organisations that are characterised by very different legal, operative and technical environments affords greater scope for comparative analysis and understanding the biography of packages. We intend to conduct participant observation research at user group conferences and feedback sessions.

This rich combination of data collection methods enables detailed current and longitudinal analysis and comparisons between cases in different sectors and stages in the package maturation. The biography approach is a novel and ambitious one: it is only because we can draw upon the distinctive historical and interdisciplinary knowledge base brought together in this research team that we are able to assemble a comprehensive and in-depth picture of the evolution of packages for the greater part of their lifecycle, from their early stages of conception to today, including projections of future developments.

Bibliography

- Appadurai, A, (1992), (Ed.), The Social Life of Things: Commodities in Cultural Perspective, Cambridge, Cambridge University Press.
- Bansler, J & Havn, E, (1996), Industrialised Information Systems Development, CTI Working Paper No. 22, Technical University of Denmark.
- Brady, T, Tierney, M & Williams, R, (1992), The Commodification of Industry Applications Software. Industrial and Corporate Change. 1(3), p489- 514;
- Ciborra, C, and associates, (2000), From Control to Drift: The Dynamics of Corporate Information Infrastructures, Oxford, Oxford University Press.
- Cornford, J & Pollock, N, (2003), Putting the University Online: Information, Technology and Organisational Change, Milton Keynes, Open University Press.
- D'Adderio, L, (2001), Crafting the Virtual Prototype: How Firms Integrate Knowledge and Capabilities Across Organisational Boundaries, Research Policy, 30: 1409-1424.
- D'Adderio, L, (in press), Inside the Virtual Product: How Organisations Create Knowledge Through Software, Cheltenham: Edward Elgar.
- Davenport, T, (1998), Putting the Enterprise into the Enterprise System. Harvard Business Review, 76(4), p.121-132
- Davenport, T, (2000), Mission Critical: Realising the Promise of Enterprise Systems, Boston, Harvard Business School Press.
- Davis, J, (1998), Scooping up Vanilla ERP: Off-the-Shelf Versus Customised Software, InfoWorld, 20(47): 1-4.
- Ellingsen, G, Monteiro, E & Moen, R, (2002), Reforming the Unreformable Process: Electronic Patient Records in Large Hospitals in Norway 1980s – 2000, Precept Work Package 3: Case Studies of Social Shaping of BPR, Precept working paper No. 29 at <http://www.its.dtu.dk/faggr/tesoc/precept/workpapers/default.htm>
- Fleck, J, (1993), Configurations: Crystallizing Contingency, Human Factors in Manufacturing, 3, (1), p. 15-36.
- Friedman, A, (1989), Computer Systems Development: History, Organisation & Implementation, Chichester, Wiley.
- Hanseth, O, Ciborra, C & Braa, K, (2001), The Control Devolution: ERP and the Side-effects of Globalisation, The Data Base for Advances in Information Systems, 32, 4, p.34-46.
- Hanseth, O & Braa, K, (2000), 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.
- Hartswood, M, Proctor, R, Rouncefield, M, Slack, Voss, A, & Williams, R, (2002), Building Information Systems as Universalised Locals, Journal of Knowledge, Technology and Policy.
- Kjaer, A & Madsen, K, (1997), Customer-Vendor Co-operation, Information Technology & People, Vol. 10, pp205-223.
- Koch, C, (1997), Production Management Systems: Bricks or Clay in the Hands of Actors in Clausen, C, & Williams, R, The Social Shaping of Computer-Aided Production Management and Computer-Integrated Manufacture, European Commission, DG X11/B.1, Luxembourg.
- Koch, C, (1999), SAP R3 - An IT Plague or the Answer to the Tailor's Dream?, Working Paper, Institute for Technology and Social Science, Technical University of Denmark.

Koch, C, (2001). BPR and ERP: Realising a Vision of Process with IT, Business Process Management Journal, Vol. 7, No. 3, pp258-265.

Light, B, (2001), The Maintenance Implications of the Customisation of ERP Software, Journal of Software Maintenance and Evolution: Research and Practice, 13, 2001, p.415-429.

McLaughlin, J, Rosen, P, Skinner, D, & Webster, A, (2000), Valuing Technology: Organisations, Culture and Change, London, Routledge.

Orlikowski, W, (1992), The Duality of Technology: Rethinking the Concept of Technology in Organizations. Organization Science,3,(3), p.398-427.

Parr, A & Shanks, G, (2000), A Model of ERP Project Implementation, Journal of Information Technology, 15, p. 289-303.

Pollock, N, (2000), The Virtual University as Timely & Accurate Information, Information, Communication & Society, 3, 3, p349-365.

Pollock, N, (2003), The 'Self-Service' Student: Building Enterprise-Wide Systems into Universities, Prometheus, Vol. 21. No. 1., p101-119.

Pollock, N, Williams, R, & Procter, R, (in press), Fitting Standard Software Packages to Non-Standard Organisations: The 'Biography' of an Enterprise-Wide System, Technology Analysis & Strategic Management.

Quintas, P, (1994), Programmed Innovation? Trajectories of Change in Software Development, Information Technology & People, Vol. 7, No. 1, pp25-47.

Sawyer, S, (2000), Packaged Software: Implications of the Differences from Custom Approaches to Software Development, European Journal of Information Systems, 9, p.47-58.

Schumm, W & Kocyba, H, (1997), Re-Contextualisation and Opportunities for Participation in Clausen, C, & Williams, R, The Social Shaping of Computer-Aided Production Management and Computer-Integrated Manufacture, European Commission, DG X11/B.1, Luxembourg.

Scott, J & Kaindl, (2000), L, Enhancing Functionality in an Enterprise Software Package, Information & Management, 37, p.111-122.

Sorensen, K, (1996), Learning Technology, Constructing Culture: Socio-technical Change as Social Learning. STS Working Paper No. 18/96, Centre for Technology & Society, Trondheim.

Scott, S, & Wagner, (in press), Networks, Negotiations, and New Times: The Implementation of Enterprise Resource Planning into an Academic Administration, Information and Organisation.

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

Webster, J & Williams, R, (1993), Mismatch and Tension: Standard Packages and Non-standard Users, in P. Quintas (Ed.) Social Dimensions of Systems Engineering (Hemel Hempstead, Ellis Horwood).

Williams, R, (1997), 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, p. 170-185.

Williams, R, Stewart, J, & Slack, R, (Eds.), (1999), The Social Shaping of Multimedia, COST A4, European Commission DGXIII, Luxembourg.