

Software and Organisations

The biography of the enterprise-wide system or how SAP conquered the world

**Neil Pollock and
Robin Williams**

Software and Organisations

The modern enterprise-wide information system has become a software package. A small number of software suppliers, of which the software giant SAP is the clear leader, have apparently succeeded in deploying their enterprise resource planning (ERP) solutions across many different organisations, sectors and countries around the globe. This illustrates a significant shift – involving the reshaping of the corporate information system.

This is the first book that addresses the genesis and career of the modern-day enterprise system in a comprehensive and robust manner. It does so through setting out a new approach for the study of packaged solutions and presents novel empirical studies based on in-depth ethnographic and longitudinal research conducted within supplier organisations and other relevant sites. The authors shift the debate within the social study of information systems, from one that is primarily focused on ‘implementation studies’ to one that follows software as it evolves, matures and crosses organisational boundaries. Through tracing and comparing the ‘biography’ of a number of software systems, the authors develop a new vocabulary for the dynamics that surround standardised software.

Original in its approach, this book draws on a number of ethnographic studies in supplier organisations, user settings, user forums, and applies theories from the Sociology of Technology, Technology Studies, Innovation Studies, and beyond. As such it will be of interest across all of these subject areas and to researchers from the wider fields of Information Systems and Business Studies.

Neil Pollock is a Senior Lecturer at the University of Edinburgh where he teaches and researches on the sociology of information technologies.

Robin Williams is Professor of Social Research on Technology and Director of the Institute for the Study of Science, Technology and Innovation at the University of Edinburgh.

Routledge Studies in Technology, Work and Organisations

Edited by David Preece, University of Teeside, UK.

1. Information Society and the Workplace

Edited by Tuula Heiskanen and Jeff Hearn

2. Technological Change and Organizational Action

Edited by David Preece and Juha Laurila

3. Managing Complex Projects

Networks, knowledge and innovation

*Neil Alderman, Chris Ivory, Ian McLouglin, Alfred Thwaites
and Roger Vaughan*

4. Information and Communication Technologies in rural society

Being rural in a digital age

Grete Rusten and Sarah Skerratt

5. Software and Organisations

The biography of the enterprise-wide system or how SAP
conquered the world

Neil Pollock and Robin Williams

Software and Organisations

The biography of the enterprise-wide system or
how SAP conquered the world

**Neil Pollock and
Robin Williams**

First published 2009
by Routledge
2 Park Square, Milton Park, Abingdon, Oxon OX14 4RN

Simultaneously published in the USA and Canada
by Routledge
270 Madison Ave, New York, NY 10016

Routledge is an imprint of the Taylor & Francis Group, an informa business

This edition published in the Taylor & Francis e-Library, 2008.

“To purchase your own copy of this or any of Taylor & Francis or Routledge’s collection of thousands of eBooks please go to www.eBookstore.tandf.co.uk”

© 2009 Neil Pollock and Robin Williams

All rights reserved. No part of this book may be reprinted or reproduced or utilised in any form or by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying and recording, or in any information storage or retrieval system, without permission in writing from the publishers.

British Library Cataloguing in Publication Data

A catalogue record for this book is available from the British Library

Library of Congress Cataloging in Publication Data

Pollock, Neil, 1967-

Software and organisations : the biography of the enterprise-wide system or how SAP conquered the world / Neil Pollock and Robin Williams.
p. cm.

Includes bibliographical references and index.

1. Computer software industry. 2. Management information systems. I. Williams, Robin. II. Title.

HD9696.63.A2P65 2008

338.47005 – dc22

2008008805

ISBN 0-203-89194-5 Master e-book ISBN

ISBN 978-0-415-40397-9 (hbk)

ISBN 978-0-203-89194-0 (ebk)

Contents

<i>List of figures and tables</i>	viii
<i>Acknowledgements</i>	ix
<i>List of abbreviations</i>	xi
Introduction: The reshaping of the modern enterprise solution	1
<i>Rhetorics of technology supply</i>	5
<i>Reaction against packaged solutions</i>	7
<i>Aim of the book</i>	10
<i>Research scope</i>	13
1 The dynamics of software packages	19
<i>The emergence and growth of the packaged software sector</i>	20
<i>Integrated enterprise systems</i>	22
<i>Research questions arising with the adoption of ERP</i>	32
<i>The design and development of ‘universal’ solutions</i>	32
<i>Selection and procurement</i>	36
<i>Implementation choices – how much customisation?</i>	41
<i>Post implementation</i>	46
<i>Maintenance, upgrades</i>	46
2 Critique of existing knowledge	51
<i>Introduction</i>	51
<i>Interrogating the promises: addressing the rhetorics of technology supply</i>	53
<i>Understanding technological innovation</i>	56
<i>Contributions from other disciplines</i>	63
<i>Comments on the uneven state of knowledge on ERP</i>	69
<i>STS search for analytical models and methodologies</i>	74
3 The Biography of Artefacts Framework	80
<i>How modes of research frame the analysis</i>	80
<i>Existing studies of technology and work organisation and their shortcomings</i>	83
<i>Arena and agora: concepts for exploring this complex space</i>	96

Multiple methods 109

Anthropological perspectives on artefacts 114

Reinterpreting ERP through a biographical lens 118

- 4 Fitting standard software to non-standard organisations** 128
- Recycling software* 128
 - How are ERP systems designed?* 129
 - Reasons for choosing an ERP package* 134
 - Becoming a pilot site* 136
 - The genesis of Campus* 138
 - There is no organisation there...* 141
 - Building a global product* 143
 - Incorporating specific contexts* 145
 - Conclusions: building a package through bootstrapping* 148
- 5 Generification work in the production of global solutions** 151
- Systems do not travel* 151
 - Narrative biases: localisation* 153
 - From importing to exporting* 154
 - The studies* 155
 - Birth of a package* 156
 - Management by community* 158
 - Management by content* 162
 - Management by social authority* 166
 - Promising future?* 169
 - Opening the black box (and finding a 'black blob')* 172
 - Conclusion: black blobs travel better than black boxes* 174
- 6 Technology choice and its performance: towards a Sociology of Software Package Procurement** 176
- Two incommensurable positions* 176
 - Grey space* 178
 - Theories of technology choice* 180
 - Arriving after the event* 184
 - Public procurement* 186
 - Bringing options to the table* 187
 - Dividing possibilities* 188
 - Narratives of success* 190
 - The provenance and status of the software packages* 191
 - Credibility contests: providing evidence of competence* 193
 - Have the experts heard of them?* 195
 - Shifting technology preferences* 198
 - 'We are duty bound to get to the facts'* 199
 - Seeing is believing* 202
 - Conclusions: comparative measures can be performed* 205

7 Industry analysts and the labour of comparison	210
<i>An alternative view on simple market tools?</i>	211
<i>The growing influence of industry analysts</i>	213
<i>How does low-status knowledge spread?</i>	214
<i>Theorising simple tools</i>	215
<i>The Magic Quadrant</i>	221
<i>A transformative tool?</i>	222
<i>Shaping the Magic Quadrant</i>	224
<i>The objectification of community knowledge</i>	227
<i>Extending the Magic Quadrant outside Gartner's offices</i>	229
<i>How satellites report back to Gartner</i>	235
<i>Conclusions: reconstituting the technical field</i>	238
8 Passing the user: searching for expertise in globalised technical support	241
<i>The organisation of technical support</i>	244
<i>Distributed forms of support</i>	246
<i>Setting and research methods</i>	248
<i>Globalised face-to-portal forms of technical support</i>	252
<i>Reconstituting the nature of technical support</i>	256
<i>Reconfiguring relationships</i>	259
<i>Raising new technologies and policies</i>	264
<i>Escalating beyond the portal</i>	266
<i>Conclusions: putting the user at the centre of the distributed organisation</i>	271
9 Discussion and conclusions	275
<i>Advancing the social analysis of technology</i>	275
<i>Conceptualising the biography of software packages</i>	281
<i>Across the software lifecycle</i>	282
<i>Across multiple product cycles</i>	289
<i>Theorising relationships at multiple levels and multiple timescales</i>	290
<i>The process of technological innovation</i>	295
<i>Commodification of networked reputation: the role of industry analysts</i>	297
<i>How SAP conquered the world?</i>	298
<i>Extending our approach</i>	301
<i>Notes</i>	303
<i>Bibliography</i>	319
<i>Index</i>	342

List of Figures and Tables

Figures

1.1	Adoption of the ERP concept in IS academe	28
3.1	Mapping the broader structure in a multi-local analysis	94
3.2	Hutchins' cube	108
3.3	Article counts of old innovation concepts and ERP	124
3.4	Evolution of ERP	126
5.1	Performing community	160
5.2	Pilot site at the lectern	160
5.3	The Campus community	170
5.4	Proximity of users to artefact	170
5.5	Generic solution as a 'black blob'	173
6.1	The demonstration	206
7.1	The calculative network	236

Tables

1.1	Davenport's 1998 list of functions contained in SAP's R/3 package	23
1.2	Current list of functions contained in SAP, R/3 package	24
1.3	Information systems publications and presentations on ERP 1997–2000	28
1.4	SAP industry-specific business maps	30
1.5	ERP market	31
1.6	Typology of ERP customisation options	45
3.1	Extended ERP	125

Acknowledgements

Conducting a biographical study of the modern enterprise-wide information system was far from an easy task. It required much support and help from many people and organisations. This included the provision of funding which provided us with the time and resources to carry out the research on which the book is based. It also include finding and then securing access to the locations in which these software solutions about which we write are designed and used. In this former respect we would like to acknowledge the generous financial support of the UK Economic and Social Research Council (ESRC) which funded our research project ‘The Biography and Evolution of Standardised Software’ (Award No. RES-000-23-0466) as well as much earlier research underpinning this enquiry. In terms of the latter, we thank the numerous people who provided us with insights into the trials and tribulations of designing, implementing and living with packaged software systems and especially the general ‘Campus’ user community. For providing us with intellectual and practical support, we are also grateful to our colleagues involved in the Biography of Software project. These were James Cornford, Luciana D’Adderio, Christine Grimm, Rob Procter, Christine Schwarz and Mei Wang. There were of course many other people who have helped along the way through either reading our drafts or providing us with valuable comments and ideas for exploration. These include Stuart Anderson, Geoffrey Bowker, Raluca Bunduchi, Lawrence Busch, Barbera Czarniawska, Wendy Faulkner, Jamie Fleck, Martina Gerst, Ian Graham, Ole Hanseth, Mark Hartswood, Stephen Harwood, Sampsa Hyysalo, Christian Koch, Irvine Lapsley, Michael Lynch, Antonios Kaniadakis, Ben Light, Ashley Lloyd, Donald MacKenzie, Eric Monteiro, Daniel Robey, Stewart Russell, Graham Spinardi, Dave Stearns, James Stewart, Alex Voss and Geoff Walsham. We also acknowledge and thank the more general Science and Technology Studies community at Edinburgh University for providing us with a convivial and stimulating environment in which to work. We particularly appreciate the input of Christian Koch and Sampsa Hyysalo who found the time, not to mention enthusiasm, to read and comment on the entire manuscript when it was still in a very early and somewhat undigested state. We are grateful also to Ben Light who provided us with

comments on the final manuscript. We thank Christine Grimm who carried out much of the fieldwork reported in Chapter 8; she also contributed to analysing this material and we have thus included her as a co-author on this particular chapter. We acknowledge the contribution of Tasos Karadedos who accompanied us during interviews at 'EduCo'. His assistance and final dissertation were very helpful in preparing Chapter 5. The research presented in Chapter 4 was funded under an ESRC project called 'Students as Users' (Award No. RES-000-22-3276). Neil Pollock would also like to thank David Charles, James Cornford, John Goddard, Kevin Robins and Frank Webster for their support during this project. The research presented in Chapter 6 was conducted while Neil Pollock was working on the 'AMASE' project. This was a project funded by the Engineering and Physical Science Research Council (EPSRC) investigating software and organisational integration in the public sector. We would like to thank those people at the local authority who generously allowed us to carry out fieldwork. Neil would also like to thank his AMASE colleagues at the University of Newcastle upon Tyne, especially Elaine Adams, James Cornford, John Dobson, Ian McLoughlin, Mike Martin, Ros Strens, Roger Vaughan and Bridgette Wessels. Finally Neil would like to thank James Cornford for his help and guidance during his time at the Centre for Urban and Regional Development Studies (CURDS) at the University of Newcastle upon Tyne. He would especially like to thank Luciana and Luca for all the support and love they gave to him while he was working on this book.

Chapter 4 is a modified version of our previously published article 'Fitting Standard Software to Non-Standard Organisations', *Technology Analysis and Strategic Management*, 15 (3) (2003), pp. 317–32. The authors were Neil Pollock, Robin Williams and Rob Procter. We would like to thank Rob Procter for his permission to reproduce parts of our joint paper. Chapter 5 was based upon an article previously published as 'Global Software and its Provenance: Generification Work in the Production of Organizational Software Packages', *Social Studies of Science*, 37 (2) (2007), pp. 254–80. The authors were Neil Pollock, Robin Williams and Luciana D'Adderio. We would like to thank Luciana D'Adderio for her permission to reprint this joint paper. Chapter 6 was published as 'Technology Choice and its Performance: Towards a Sociology of Software Package Procurement', *Information and Organization*, 17 (2007), pp. 131–61. The authors were Neil Pollock and Robin Williams. We are grateful to the copyright holders for permission to use this material here. We also thank those who permitted us to reproduce figures and photographs. In particular, these were Tasos Karadedos, Annemie Depuydy and Ping Wang.

List of Abbreviations

ACME	Application of Computers in Manufacturing and Engineering
ANT	Actor Network Theory
API	Application Program Interfaces
APICS	American Production and Inventory Control Society
BCG	Boston Consulting Group
BPICS	British Production and Inventory Control Society
BPR	Business Process Redesign
CAPM	Computer-Aided Production Management
CIM	Computer-Integrated Manufacture
COPICS	Communications Oriented Production Information and Control System
COTS	Commercial Off-The-Shelf
CRM	Customer Relationship Management
CSC	Company Social Constitution
EAI	Enterprise Application Integration
ERP	Enterprise Resource Planning
EPSRC	Engineering and Physical Sciences Research Council
ESRC	Economic and Social Research Council
FTSE	Financial Times Stock Exchange
HER	Higher Education and Research
HR	Human Resources
IBM	International Business Machines Corporation
ICD	International Classification of Diseases
ICT	Information and Communication Technology
IS	Information Systems
IT	Information Technology
JIT	Just in Time
MAC	Management Administrative Computing
MIT	Massachusetts Institute of Technology
MRP	Materials Requirements Planning
MRP II	Manufacturing Resource Planning
MQ	Magic Quadrant
OECD	Organisation for Economic Co-operation and Development

OHP	Overhead projector
OJEC	Official Journal of the European Communities
OSI	Open Systems Interconnection
PDM	Product Data Manager
PhD	Philosophy Doctorate
PICT	Programme in Information and Communication Technology
POC	Proof of Concept
RE	Requirements Engineering
REM	Real Estate Module
RfI	Request for Information
RfP	Request for Proposals
SaaS	Software as a Service
SAP	Systems, Applications, Products in Data Processing (SAP AG)
SCM	Supply-Chain Management
SCOT	Social Construction of Technology
SERC	Science and Engineering Research Council
SIIA	Software and Information Industry Association
SIS	Student Information System
SME	Small and Medium-sized Enterprises
SSK	Sociology of Scientific Knowledge
SST	Social Shaping of Technology
STS	Science and Technology Studies
TCE	Transaction Cost Economics
TEM	Training and Events Module
TQM	Total Quality Management
UCAS	Universities and Colleges Admissions Service
VAR	Value-Added Reseller
XML	eXtensive Markup Language
Y2K	Year 2000

Introduction

The reshaping of the modern enterprise solution

November 1990: The Hague. *The UK Science and Engineering Research Council (SERC) sent one of its leading experts to an international workshop organised to discuss the future of the computer systems used to run industrial enterprises. The workshop was one of a number organised in Europe and the USA that year to assess the prospects for these technologies that were seen as constituting 'best practice' in manufacturing organisations and crucial for industrial competitiveness.*

The workshop, organised by the Eindhoven Group, widely regarded as the 'leading research group in Europe' on these technologies, attracted a strong and interdisciplinary turnout, with over sixty consultants, technology vendors, users and academics coming together to discuss its provocative 'rationale document'. Gerry Waterlow (consultant to SERC's Application of Computers and Manufacturing and Engineering Directorate) circulated a report, drawing attention to the consensus that appeared to have been reached around the central argument advanced by this document. He suggested that these conclusions could probably be regarded as a 'reasonable snapshot' of the direction in which the particular technology they were all there to discuss was moving. It was even suggested that – such was the consensus – the workshop itself might help to underwrite this future direction since many of the actors central to its shaping were present in the room.

The technology under the microscope was the state of the art of what we today would call Enterprise Resource Planning solutions – known then as Manufacturing Resource Planning (MRP) systems. In the late 1980s, MRP technology had been heavily promoted as a solution suitable for a wide range of organisations. However the title of the workshop – 'Beyond MRP: MRP and the Future of Standard Software for Production Planning and Control' – made it clear something was afoot. One did not have to read too far into the workshop rationale document to see the sting:

The development of MRP (I and II) has led...to a specific production control philosophy [as well] as to standard software for production control. Control philosophy and standard software are heavily intertwined. Having standard software for production control is very important in

2 Introduction

practice, as well with respect to the whole implementation and training process as with respect to maintenance. On the other hand MRP (philosophy and software) seems not to fit well everywhere.

(Workshop Rationale Document).

The workshop had been motivated by growing concerns that this latest breed of enterprise system was proving problematic. Users, it seems, found these systems 'difficult to apply' and as a result, they were 'not widely adopted'. Some of the difficulties experienced concerned their 'generic' nature and it was generally perceived that the processes embedded in the software were 'too rigid' for most adopting organisations. Indeed, the Eindhoven Group saw the workshop as a means to 'debate the reasons' as to why this was, as well as to 'identify ways forwards'. They concluded that it was 'time to discuss the future of standard software in general and more specifically the future of MRP'. The workshop put forward 'three scenarios for MRP development:

- 1 gradual evolution of generalised MRP with the existing software suppliers remaining the major vendors;*
- 2 increase in user-driven special versions of MRP for particular industries, leading to partnerships between users and smaller suppliers concentrating on vertical markets;*
- 3 decline in significance of MRP [to be replaced by] factory management systems, supplied by system integrators with a broad range of skills (systems, software, communications, automation) [which] will take over MRP II functions, (Waterlow 1990: 2).*

The conference background paper (Wijngaard 1990: 5) described the latter as the 'more radical scenario, and one for which there is substantial evidence that new ideas are emerging from outside the MRP world'. Concepts such as Just-In-Time and Computer-Integrated-Manufacture would, it argued, be 'captured better by factory management systems rather than MRP'. Moreover:

[t]he special needs of industry sectors cannot be met by a generic MRP system, and different methods will emerge. These developments are being made today partly by a new group of 'systems integrators' who have stronger technical skills in systems, communication, automation, and new software technologies. In this scenario the structure of the software industry is likely to change as new suppliers appear.

(ibid.: 5)

During the workshop, there was the general feeling that the current direction of the enterprise system was not sustainable. 'The majority [present at the conference] considered MRP2 in the form of standard software as an unworkable concept' (ibid.: 4). The future as they saw it was not with generic

software packages but instead there was an 'urgent need' for alternative more 'context specific software packages' (ibid.: 6). Intriguingly, the workshop deliberations showed little awareness that waiting just around the corner was a new breed of software supplier that would herald in a very different future for the enterprise system...

The modern enterprise-wide information system *has* become a generic software package. A small number of software suppliers, it seems, of which the German-based software company SAP is the clear leader, have succeeded in deploying their Enterprise Resource Planning (ERP) systems within and across many different organisations, sectors and countries around the globe. Large corporations and organisations throughout the world now appear to be dominated by a new breed of standardised packaged solution. How has this happened? Indeed, the fact it has happened at all is remarkable when one considers that only two decades ago leading experts and practitioners agreed that the future for organisational information systems was *not* with generic IT solutions. Back then, and based on experience with the state-of-the-art integrated enterprise planning solutions of the day, many experts considered it highly unlikely that a small number of generic information systems could meet the needs of organisations within and across sectors (Waterlow 1990). These systems were seen as *too* standardised for the complex and diverse needs of adopting organisations (which, as they saw it, required alternative and more flexible, locally specific kinds of solutions). Thus, the future for technology supply was seen to lie with vendors developing varieties of sector-specific offerings that could be locally adapted to the various particular user organisations seeking to apply them. Discussions of sectoral difference and organisational uniqueness were the order of the day and 'semi-generic' and highly tailorable packages were seen to be the way forward.

However, while the 'structure of the software industry' has changed this is not in the way predicted by the Eindhoven Group. Despite their assessment at the 1990s workshop, a new breed of software package supplier has emerged which has managed to *reuse* and *recycle* highly standardised systems into thousands of different organisations. These packaged solutions now make up a substantial part, perhaps the majority of organisational IT expenditure (Jakovljevic 2001) and include, as well as ERP, Customer Relationship Management (CRM), Supply-Chain Management (SCM) and other financial and administrative systems. The extension of the generic package into organisations worldwide is all the more remarkable when one considers that not only was the phenomenal success of suppliers like SAP 'not on the radar' during the workshop, but the vision for the direction of these systems was far removed from what we have today. However, before the end of that decade, SAP's now famous 'R/3' package, followed by other suppliers of similar generic software solutions that have become known as Enterprise Resource Planning systems, would have swept through major corporations in Europe, the USA and beyond, moving out from manufacturing into services and the public sector. This poses the following

4 Introduction

questions: How has this happened? How has this new breed of supplier been able to extend their systems into organisations worldwide? Moreover, what does this mean for the character of the organisational information system and wider arena in which they are situated?

What is clear from the rise of these kinds of solutions is that the nature of the modern enterprise system is changing. Not only have these new suppliers recycled their technologies into many different places but, arguably, in doing so, they have heralded a shift in the conception of the organisational information system. What is at stake is a profound change in ideas about the very notion of the modern corporate solution: this encompasses how they should be developed and implemented as well as the extent to which they should address particular sectoral and organisational requirements. Clearly, these new kinds of systems have important implications for researchers interested in the technology and organisation relationship. How are we to respond to the rise of this new breed of software supplier and the extension of the generic enterprise system in a sensible and comprehensive way (i.e. without either inflating or reducing these changes)?

However, despite the fact these kinds of standardised packaged solutions account for the bulk of systems used today, we cannot in a conceptually and empirically robust manner explain their rise to prominence. We do not know precisely *how* the modern corporate system became a generic package. Though practitioners may advance well-rehearsed ‘potted histories’ of these artefacts there are very few studies of their origination and design, let alone research which addresses the evolution of this technology along its protracted lifecycle. These kinds of IT systems have had nowhere near the kind of sociological attention they deserve. Why is this?

One reason is that the received wisdom among many scholars interested in the social study of technology would be that generic solutions have only limited applicability: for some, there is no such thing as a ‘universal’ or ‘one size fits all’ solution (Star and Ruhleder 1996; Hanseth and Braa 2001). Standard systems only work to the extent they are adapted by user organisations through messy localisation processes. Thus, according to many sociologists it is ‘users’ and ‘adopting organisations’ that should be studied. While, on the one hand, we share their interests in implementation, it has also meant that, on the other, recent research on information systems has become somewhat unbalanced. In focusing principally upon user organisations, social scientists have not adequately conceptualised and analysed standardised information solutions. There is not, for instance, a comprehensive understanding of the inner workings of the leading software supplier organisations. Nor do we have a sophisticated appreciation of the wider information system industry dynamics that surround software producers. This is reflected by the lack of frameworks that explain the extension of these systems across sectors and this wider shift from specialised to generic software. Let us briefly look at some of the dominant ways these technologies are researched.

The current social science research on packaged enterprise systems is broadly gathered around two opposing poles. The first, typified by more managerially focused kinds of analysis, views ERP systems and the like as more or less ‘transformatory’ technologies containing ‘universal logics’. They imply that because of the nature of their design these systems can be applied extensively across all kinds of corporations and bring about widespread change (see for instance O’Leary [2000] and Bendoly and Jacobs [2005]). Not surprisingly, this view has been seen as problematic by critical social scientists. Thus, a second pole has subjected these discourses of transformation and universalism to critical assessment. In what might broadly be characterised as the ‘Social Study of Information Systems’ (McLaughlin *et al.* 1999; Ciborra 2000; Walsham 2001; Sawyer 2000; Avgerou 2002) scholars have advanced alternative accounts of the spread of these solutions. Many have produced what might be described as ‘situated’ and ‘localist’ explanations, often drawing on the groundbreaking work of Suchman (1987) as well as ethnographic study (see for instance Knox *et al.* [2005]). These accounts typically contrast the uniqueness in structure and practices of user organisations with the standardisation of packaged solutions, and have tended to emphasise the ‘contingency’ surrounding the implementation of these systems (Hanseth *et al.* 2001).

However, while this literature is highly informative, it also tells us rather little about what we regard as one of the most important developments in the short history of corporate information systems: the shift from locally specific to generic systems. Scholars in the Social Study of Information Systems, for instance, have thus focused selectively upon certain aspects and moments of the software package lifecycle and as a result, they offer what have now become well-rehearsed but also highly partial accounts. Critical social science should be able to give a more comprehensive analysis of the reshaping of the modern corporate system. Not to do so has risks – handing the terrain to other disciplines.¹ This has meant that the debate around enterprise systems has been unevenly developed and unhelpfully fragmented between rather narrow (e.g. managerial or technical) perspectives. Of course, every failing is also an opportunity, and the gap that exists in our current understanding is also one we hope to fill (at least in part) with this book.

Rhetorics of technology supply

Today, few can deny that packaged highly standardised forms of enterprise solutions have become an important feature of our organisational landscape. In this respect, Management scholars have been prolific in celebrating their various features and characteristics. Daniel O’Leary (2000), for instance, goes as far as describing systems like ERP as nothing less than a ‘corporate marvel’. They have undoubtedly had an enormous influence on the business and information system worlds, he argues, affecting each of the following dimensions. They have experienced a huge market growth, being taken up by most

6 Introduction

of the major corporations around the world. They are also now increasingly being rolled out within small and medium-sized enterprises. Moreover, within corporations they have been used as one of the primary tools for re-engineering the organisation as well as the diffusion of many best practices.

Added to this, there is also the (mostly implicit) assumption that they have heralded in a new class of computer solution (Klaus *et al.* 2000). This, first, is the suggestion that the ‘generic-ness’ of these solutions is an achievable design issue (Carey 1998). In addition, that these solutions can be ‘recycled’ across similar classes of organisations (Deifel 1999). This can be within the same or related industrial sector or, as is now increasingly common, across different and unrelated sectors and organisational forms. Second, and in stark contrast to the organisational information systems that went before, these systems are now generally thought to behave like ‘products’ that can be selected and purchased, as with other kinds of commodities (Deifel 1999; Heiskanen *et al.* 2000; Regnell *et al.* 2001; Xu and Brinkkemper 2005).

There are, of course, aspects of this brief account that deserve to be challenged, one of which is that enterprise systems were not simply born ‘software packages’ or ‘generic solutions’. Rather, something had to be done to them to achieve their ‘generic-ness’ and ‘commodity’ status. It is notable that Management research provides very little, whether in terms of empirical findings or conceptual frameworks, that will afford an adequate understanding of this ‘something’. This is true among even the more highly regarded works – such as Davenport (2000b). In terms of the first point (‘generic-ness’) this literature tells us almost nothing about how the suppliers design and develop these systems and products or on what they base the design of generic solutions. We understand very little about 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 re-designed from a generic to a niche specific solution, or whether it is being recycled from one sector to another or ‘upgraded’ from custom-built software to a generic system. In terms of the second point (‘commodity status’) we have little understanding of just how software packages are typically presented to potential adopters. This includes the different strategies and decision-making processes of those adopting software packages (the process by which users assess and make sense of the wide range of alternatives and options available). This is whether to procure one of the more generic systems on offer or a more flexible ERP alternative. On one hand, it is acknowledged that organisations find it difficult to critically assess and evaluate the range of packages on offer. Moreover, there is 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. Other considerations and actors are obviously at work here.

One of the other problems of the Management literature (including Davenport) is that it tends to be based on a particularly weak empirical base.

Rather than study actual technologies, these writers tend simply to align themselves with the statements and rhetorics of technology supply. We therefore turn to other disciplines within the social sciences, where these criticisms apply less, where there are numerous frameworks available to trace this development (although none appear sufficiently fine-tuned to analyse the ‘biography’ of the generic solution in the way we think necessary) and where, over recent years, an enormous amount of qualitative and particularly ethnographic research data has been amassed.

Reaction against packaged solutions

The Social Study of Information Systems is made up of work from scholars within Science and Technology Studies (STS), Information Systems (IS) research and Organisation Studies.² In the face of often quite deterministic and supplier-dominated debates, researchers from these approaches were among the first to characterise the complexities and difficulties associated with modern packaged information systems (see Lucas *et al.* 1988; Webster and Williams 1993; Salzman and Rosenthal 1994). Much of this ‘critical project’ has grown up in opposition to the more dominant supply-side accounts. Thus it is no surprise this work predominately focuses on the struggles adopting organisations engage in while attempting to make generic and standard systems work within their user settings. There have been many studies now showing how packaged systems seldom translate (or translate easily) across boundaries, whether these are between organisations within the same sector, between industrial sectors, or between public and private sector organisations (Pollock and Cornford 2004). The difficulties in developing solutions that can be widely applied result, it is commonly argued, from the diversity of organisational settings and the resultant gulf that exists between the system and the specific contexts, practices and requirements of particular user organisations. Indeed, if generic packages *do* work across settings, this would, under these perspectives, be seen to be only at great expense to the adopting organisation (in terms of adapting the package and prejudicing the benefits of standard solutions or imposing unwanted organisational change in order to meet presumptions built into the package). Indeed, there is now a large (and rather interesting) literature on cases of failure, implementation difficulties, and on the costs and risks associated with adopting these systems (Scott and Vessey 2002; Newman and Westrup 2005; Wagner and Newell 2006).

Yet, if we are to view this literature a little more critically, when reading some of these studies it is a wonder these systems extend at all. The latter sets of arguments are pursued with such vigour – there is such a desire, it seems, to demonstrate the complex organisational and technical reworkings necessary to sustain packaged software – that there appears to be an entrenched scepticism with regard to their wider applicability (Soh *et al.* 2000; Scott and Wagner 2003; Soh and Sia 2004). For many social scientists,