Preface

INTRODUCTION

The purpose of this book is to bring together empirical and theoretical research on adaptive and evolutionary information systems (IS) from researchers and practitioners. The subjects of evolution and adaptation have been taken together because they describe aspects of the same phenomenon. The phenomenon is the use of information technology (IT) in context and situation or in the system's environment. In all evolutionary and adaptive IS and IT, time is the unique feature of evolution. Systems need to adapt to their environment, usually in a specific context, and often for a specific situation over time. Whilst systems evolution may be regarded as the transformation of a system to reflect its environment, adaptation is the software mechanism by which such evolution occurs. Context and situation are unique features of adaptation and evolution.

Concepts like context, situation, interaction, evolution, and adaptation are important for computer scientists and IS researchers and practitioners. They contribute to building systems that reflect their environments. It is becoming increasingly apparent that the domain of application and its environment is in all respects dynamic. Individuals' and groups' needs, team workflows, business processes, and business strategies all are changeable. Whether the application is in business, government, or healthcare, the inherent nature of the domain and environment is dynamic. Change is all around.

Adaptation is concerned with analysing and understanding the interface, or more accurately the interaction, between human behaviour and computer systems, especially in the form of IS. The aim of researchers and practitioners contributing to this book is to improve our knowledge of the principles involved in the generation of evolutionary and adaptive behaviour in IS. Some researchers seek to develop system models reflecting the contingencies of adaptive behavioural responses in adaptive systems. The challenge that faces all researchers in this field is the development of adaptive systems for unpredictable and unforgiving application domains.

From a political perspective, the notion of systems evolution is critical to the development of IS that can reflect the nature of modern liberal democracies. IS designs cannot be deterministic in our modern information society. IS designs that lack adaptability are less desirable in market economies. So the evolutionary perspective on IS taken in this book is important as the freedoms to think, communicate and act are values worth reflecting in technologies. Whilst governments may try to deny these in times of perceived threat, the development in IT has undermined their attempts to control citizens' activities. The Internet and the World Wide Web recognise no censorship, although this is now changing. Whatever one's politics, IT has firmly established itself in the 21st century society. Our conceptions and explanations of IS, their development and, more importantly from the standpoint of this book, their usage need to recognise the power of technology to support humans to express themselves, whether in business, government, healthcare or in society generally.

EVOLUTIONARY AND ADAPTIVE SYSTEMS

The subject of evolutionary computer systems is traceable to Lehman (1980) commenting on program evolution. Since then many models of business such as competitive models and business process models have influenced the application of IT to business. These have resulted in process-based concepts of evolutionary IS in business (Kawalek & Leonard, 1996; Warboys, 1991; Warboys, 1994; Warboys, Kawalek, et al., 1999). The need for evolutionary systems is now greater in business. Drucker (1993, 1998) posits that the 21st century business organisation is subject to changing economic and competitive pressures, flatter internal organisation structure, and new organisational forms.

A system is evolutionary if it can be *purposefully used* in a dynamic environment. Human purposes are changeable, whether they be the purposes of an individual, group, company, or government or healthcare organisation. An example of an evolutionary IS is a medical diagnostic system in an environment of a changing population, or a local government system for child benefit in an environment of changing legislation, or a strategic online airline reservation system in an environment of lower revenues.

There are many types of evolutionary systems. Smart evolutionary systems use artificial intelligence methods or intelligent agents, where the system learns to adapt by itself learning how to respond to environmental change. Other evolutionary systems are those that are adapted by people in specific situations. Adaptive behaviour may be regarded as that behaviour which aims at removing or circumventing a situation where a goal is blocked. We are interested in IS that poses these and other adaptive properties.

COMPLEX SYSTEMS ENVIRONMENTS

We have come to a period in IS where the past 40 years work needs to be re-thought. Invoking a single theory to explain the various phenomena surrounding evolutionary systems in particular organizations can numb our analytical faculties. No such universals for the problem of developing evolutionary and adaptive systems are provided in this book. Instead, computer science and IS researchers need to accept that the postmodernist view (Bjørn-Anderson,1988) has relevance to addressing the problem. Though it too has to be tempered. As the postmodernist philosopher Jean-François Lyotard argues, in the postmodern period everyone is skeptical of grand theories and especially the one that explains this skepticism. The general field of IS development can benefit from taking an evolutionary perspective, and methods and methodologies need to be refined to account for dynamic aspects of the IS environment.

Researchers and practitioners need an adequate response to the problem of complexity in the system's environment. Complexity theorists tell us that freedom emerges when self-organising systems are placed between competing orders. What is the relation between evolution, adaptation and complexity? Complex evolutionary systems are those that generate order in the face of apparent chaos. They enable creativity to be maxmised and enable the most alternatives to emerge. Is it true to say that the antecedent of IS is organisation, or is there a more complex interplay between IS and organisation? The Internet or virtual organization is rightly characterized as organic, networked and emergent. But there is still much to learn about it. Bad theory or dubious design principles cannot be eradicated solely by criticism; a credible alternative needs to be offered. In this collection of work, the authors offer their contributions.

Whereas once our understanding of IS was intricately tied with our understanding of bricks and mortar organisations, it now demands to be understood in terms of clicks and mortar. IS researchers' task is to clearly state the contribution to specific issues in addressing complexity in the system environment and how evolutionary approaches can be used. In particular, they need to answer how they could be applied to solve particular problems in business, government or healthcare.

A critical place to start is the formulation of the problem itself. Much of the existing work frames the problem as engineering, specifically as software process engineering. It is stated as a problem that needs to be solved. We need to view the problem more humbly as Robinson, Hall, et al. (1998) suggests. Software is an entirely different material from steel. The properties and capacities we demand of it are, for example, robustness, versatility, resilience or agility, and any other attribute that humans choose to demand of it—the essence of evolutionary systems.

A second facet is system design and development decision-making. System adaptation is certainly open to design decisions being made by individuals and groups who make use of systems in their work. Whilst Alexandrian patterns have been adopted to improve communication among software developers, Alexander's original purpose in proposing architectural patterns was to enable inhabitants (users) of a building themselves to make design decisions (Borchers, 2001). Allowing individuals to make system design decisions is a prominent feature of tailorable computer systems and deferred system's design (Patel, 2001).

Many of the chapters in the book are concerned with developing evolutionary and adaptive IS for business domains. Businesses need to be agile and adapt to new markets and economic conditions. They need to meet new challenges and develop the opportunities they find. All this change has to be undertaken with minimal disturbance to the customer-facing business operations that provide the main revenue stream. Companies' drive to take advantage of e-business opportunities is now more realistic after the euphoria of the dot.com era, but the very same opportunities create new challenges too. The digital economy is by definition dynamic, allowing customers to digitally switch to other companies they think will offer a better service.

ORGANISATION OF THE BOOK

The contributors to this book come from computer science and IS academic disciplines, from industry and from European countries and the USA. They do not speak with a single voice. The research and practice covered in this book reflect the principles of interaction and organisation of human computer interaction, models of human behaviour, and system architecture among others. The chapters themselves cover a spectrum of IS issues and digital technology. Development methods and software paradigms, types of software technology, software and network architectures, and conceptions of evolution and adaptation are some. This diversity must be framed in terms of IS design. Thus the body of work composing this book is presented in terms of the IS design continuum formulated by Patel (Chapter I). The IS Design Continuum seeks to provide an understanding of not only how we conceive, analyse, design, and build complex software, but how that software needs to reflect its environment. The work seeks to move IS development and its use towards the social world of action in the IS Design Continnum, where evolution of systems is a necessity, the aim being to develop action-based IS and a recognition of so-called users as action developers through deferred system's design.

Part I of the book focuses on system tailoring, software paradigms, and deferred system's design. Our current view of information systems developers as professionals is examined by Patel (Chapter I). He argues for the recognition of action developers through deferred design decisions (DDD). Research into IS development revealed a dynamism in the development environment and a need to capture contextual details. He proposed the concept of IT as tailorable information systems that brings together technical and business issues.

Patel (Chapter 1) provides a meta-analysis of IS development research and practice. He classifies systems along an IS Design Continuum that maps machine-based IS design at one end and action-based IS design at the other. The mapping is in terms of various dimensions of development such as problem domain and environment, developer, and origin and time of design and implementation. This mapping results in three classes of systems design: autonomous system's design, specified system's design, and deferred system's design. The range is from machine-oriented to human-oriented design. He poses deferred system's design as providing an approach for evolutionary and adaptive system development.

Mørch (Chapter II) presents an approach for developing tailorable information systems. Tailorability is seen as a form of evolutionary application development. In the first part of the chapter he explains evolutionary processes by the principles and mechanisms of natural evolution and evolution of artifacts. The focus in the second part is on supporting evolutionary growth and control in software applications, seen from the perspective of users participating in development. Two concepts are introduced to aid this: the resemblance relation and component aspects. These two concepts are based on similar concepts borrowed from natural and artifact evolution. The resemblance relation is realised as a user-oriented adaptation of the object-oriented inheritance relation, and component aspects is a technique for turning components "on" and "off" during development to reduce complexity. Both techniques are elements of a research agenda for future work on user-tailorable component-based information systems.

Mørch's work on evolutionary growth and control in user tailorable systems can be placed at the far end of Patel's (Chapter I) IS Design Continuum as reflecting work that seeks to develop action-based IS for the social world of action. He covers all the aspects of the IS Design Continuum but focuses on the developer, IS design activity, role of the user interface, method of design, and the ontological basis of design activity. Mørch's work is interdisciplinary as he draws on biological evolutionary theory.

Perrey and Lycett (Chapter III) examine what evolutionary and adaptive systems design can learn from history. In particular they focus on software engineering and paradigms for developing software. Through an evaluation of the characteristics of objects, components and agents, they discuss the advantages or limitations of each paradigm based on its ability to serve the purpose of building adaptive, evolutionary systems. In considering the evolution of each paradigm, the work highlights a recurrent limitation in that the development of each has been hindered by a preoccupation with addressing the shortcomings of the previous paradigm's implementation. This observation is subsequently used to recommend a line of development designed to avoid such a pitfall for adaptive, evolutionary systems.

Perrey and Lycett propose a framework to facilitate an evaluation of software development paradigms for developing evolutionary systems. Their work examines the problem domain and environment but focuses on the IS design activity aspect of the IS Design Continuum. Their work reflects the social world of action, both of developers and users of digital technology, and seeks to develop action-based IS.

Stamoulis et al. (Chapter IV) argue that a closer look at systems development methodology reveals a key obstacle for achieving information systems that are actually flexible. Any information system can only be as flexible and adaptive as it was designed to be during its development phase because systems are built to an exact, predetermined specification. To circumvent this inherent problem of methodical information systems development they propose tailorable systems based on component software. Since the needs of end users and, subsequently, the required system functionality are constantly changing, they empower the end user to tailor the behaviour of their ATOMA system at runtime, without disrupting system operation. The system itself allows for the modification of its behaviour to accommodate for new functionality demands without necessarily discarding its past behaviour. In that way, the decisions made at design time during development are no longer an impediment for its adaptivity and evolution because only a system blueprint is initially being implemented. Upon this, end users can add, delete or modify the behaviour of the system by means of (new) roles that model context dependent variations.

Stamoulis et al.'s work demonstrates how object-oriented techniques can be used to develop action-based IS. They make use of deferred design decisions (Patel, 1999) to develop tailorable mechanisms. Deferred design decisions (DDD) allow developers to enable people who use systems to embed meanings found in the context of systems, thus providing a direct relation between a system and its environment. Deferred design decisions dissipate the boundary between the digital system and the social world of human action. Their work examines the ontological basis of design activity and addresses the issue of the emergent nature of IS in the social world of action in terms of the IS design continuum. They address how action-based IS can be developed that reflect the meaning that people attach to their action. In particular, they examine whom we can consider to be a developer, origin and time of design and implementation, and method of design.

Mørch (Chapter V) proposes a way of bridging the gap between using an application and programming it with tailoring tools. Aspects and software components act as techniques that enable end-user tailoring through the user interface. This is referred to as an application unit, and everyday artifacts like a card key motivate application units. Mørch illustrates application units and their tailoring or programming by users with the BasicDraw system. Action developers, the time and origin of design decisions, and the user interface are issues addressed by Mørch in terms of the IS Design Continuum.

Patel's meta-analysis of extant deferred systems (Chapter VI) reveals the logic of deferring the IS design process to action developers, especially for IS that need to function in a complex system environment. The logic of deferring the IS design process to so-called "users" is embedded in the systems environment (SE). The thesis of this chapter is that to build true adaptable IS it is necessary to defer the IS design process to people who make use of the IS in context or the SE. The SE of IS is thus posed as the critical problem that challenges developers of adaptable software in business applications. Mechanisms for IS adaptation can be understood better if the source for the need for adaptation itself, the SE, can be modeled to identify issues that need to be reflected in adaptable IS.

Patel argues that the system-centric view of IS has meant that the study of the SE has been neglected. Therefore, conceptions of adaptable IS and approaches for developing adaptable IS have not been considered from the critical perspective of the SE. The SE has not been previously modelled or treated systematically or analytically in the system-centric paradigm literature to inform development of an IS that is responsive to its environment. The SE is clearly the subject of analysis in this chapter in terms of the IS Design Continuum.

In Part II of the book we move on to understanding the crucial need for adaptive and evolutionary systems for the Internet and telecommunications industry. Personalisation of content seems to be the direction that Web services are taking, with intelligent agents acting on behalf of surfers. Dittrich and Lindeberg (Chapter VII) are firmly placed at the social world of action end of the IS Design Continuum. Their work examines the use of meta-modelling technology for developing industrial systems that can be adopted in the future. This work is particularly important for understanding the many aspects of the IS Design Continuum. It addresses the ontological basis of design activity for the social world of action. Rather than design a "solution" for a "problem," with their industrial telecommunications and software house partners they sought to "design-for-change." They also examine the method of design to allow tailoring. They report work on a prototype and actual system that sought to develop software that was designed for specific change that is inherent in the problem domain.

McCann (Chapter VIII) considers how adaptive systems are beginning to be used in Internet-based applications to improve perceived performance. The nature of the Internet itself is such that its bandwidth fluctuates and is generally unpredictable. One method of getting performance out of the network has been to reserve resources from the client to the server. However this is a greedy solution and does not scale well. Consequently, adaptivity is becoming a popular alternative approach to the provision of Quality of Service (QoS) support. She examines how adaptive systems work in Internet applications and introduces an adaptive audio delivery system, called Kendra, which constantly revises its data delivery mechanism according to perceived network performance between the client and server; aiming to provide the best QoS possible under changing conditions. In the IS Design Continuum, her work addresses the action-based aspects of the IS design continuum.

Ghinea and Thomas (Chapter IX) explore the use of adaptive communication protocols and architecture for multimedia data transmission. They focus on the design and implementation of an adaptable protocol for the delivery of an optimal level of Quality of Perception for users of multimedia systems. Similar to the previous chapter they seek to develop an adaptable architecture that adapts to varying network conditions to maintain Quality of Service (QoS) for users. They comment on distributed multimedia QoS context that calls for adaptable protocols capable of modifying their execution pattern to suit changing environment. In their chapter, Ghinea and Thomas explore the use of adaptive communication protocols and architectures for multimedia data transmission. In particular, they touch upon their research in the area which focused on the design and implementation of an adaptable protocol for the delivery of an optimal level of Quality of Perception—a term encompassing the user aspect of the multimedia experience.

In terms of the IS Design Continuum, their work reflects on change in perception of networking researchers, as they focus on the ontological aspects of the end-user experiences. Similar to McCann's work, Ghinea and Thomas' work may be placed in the action-based IS end of the continuum as the aim of the research is "user experience."

Dron et al. (Chapter X) examine the problems that learners face in using the Web. He reports on the CoFIND Web-based work that sought to support learners using the Web. CoFIND design reflects evolutionary system design principles, and the system organises itself based on user actions to produce "emergent order." Dron's work is thus firmly placed on the action-based IS end of the IS Design Continuum, examining the ontological basis of design activity, and where the "developer" is the users' combined actions to produce emergent properties of the system. The chapter covers the origin and time of design and implementation issues of the IS design continuum.

Kostkova and McCann (Chapter XI) present the MAGNET system that enables mobile applications to adapt to changing environments in which they are used. The MAGNET system addresses the problem of availability of dynamically updated location-aware information. They present a framework for user-customised dynamic trading of services supporting runtime adaptation, and quality-of-service-based resource description. They introduce the matching function as an alternate way of storing and matching information to location-dependent mobile device usage. In terms of the IS design continuum their work furthers the understanding of the problem domain and environment and the origin and time of design and implementation.

Vassiliou et al. (Chapter XII) discuss how personalization is a means for Web site adaptation. They provide a categorization of personalization techniques and identify factors that can lead to successful Web site adaptation with personalization techniques. They propose a unified approach that integrates the categories of personalization techniques and express the importance of evaluating the automated sites for effectiveness, especially in terms of success factors. They propose the modification of implemented personalization algorithms such that they can be adapted by accepting evaluation results. In terms of the IS design continuum their work is focused on the system environment and problem domain.

Part III of the book focuses on research and practice in business, particularly at operational and strategic levels. The two fundamental premises of research and practice in the area are that IS adaptability is dependent on business strategy and that adaptability is constrained by available technology. The business organization is a particular problem for IS developers because it has aspects that are permanent, a board of directors; aspects that are semipermanent, departments; and aspects that continuously change, projects and group working. These aspects often pose barriers to designing adaptable software.

Dyer's (Chapter XIII) work is a practitioner's contribution, reflecting on conceptual and empirical research on adaptation in IS. He addresses the issue of assessing software flexibility in business applications. There is a growing requirement for the development of tools and techniques that measure the flexibility of software systems. The Software Flexibility Framework is proposed as a technique that measures the flexibility of software systems at the strategic, functional and user-oriented levels of a business. The chapter describes how the technique was designed, how it works, and how it may be applied to different business situations.

It should appeal to a wider audience than practitioners as it provides a generic framework. In terms of the IS Design Continuum, Dyer's work considers how we define the problem domain and environment, IS design activity, and method of design. In particular, Dyer considers whether it is possible to identify factors in the problem domain which influence the flexibility of software systems such that they can be used to assess the degree of flexibility of an application.

Eardley et al. (Chapter XIV) too examine flexibility in IS but considers it from a strategic perspective. Eardley's work considers the problem domain and environment and is firmly placed in the social world of action and the development of action-based IS. He examines a framework encompassing development methods that will produce more flexible strategic IS/IT applications. The work examines the problem domain and environment in terms of the IS Design Continuum. In some respects, the framework examines the ontological basis of strategic IS design activity.

Masthoff's work (Chapter XV) is an empirical evaluation of adaptive systems. She argues that evaluation needs to be an integral aspect of research into adaptive systems and discusses the issues and limitations in conducting empirical evaluations. She provides a case of empirical evaluation to show how such difficulties can be overcome. In terms of the IS design continuum, Masthoff's work contributes to the ontological understanding of adaptation.

We hope the book makes a valuable contribution for researchers and practitioners in the field. A Web site with the authors' details can be found at *www.brunel.ac.uk/~csstnvp/*. Our overall aim is to significantly improve current practice in adaptive and evolutionary IS development and to extend the range of applications that can feasibly be developed. The aim of researchers and practitioners should be to arrive at a position where a person is able to define his or her own interaction with a system over time.

Nandish V. Patel

REFERENCES

Børjn-Anderson, B. (1988). A Postmodernistic Essay on Technology Assessment. In B. Børjn-Anderson and G. B. Davies (Eds.), *Information System Assessment: Issues and Challenges*. Amsterdam: North-Holland.

Borchers, J. O. (2001). A pattern approach to interaction design. *AI & Society, 15*(4), 359-376.

Drucker, P. (1993). *Managing in Turbulent Times.* Oxford, England: Butterworth Heinemann.

Drucker, P. (1998). *The Coming of the New Organisation*. Boston: HBSP.

Kawalek, P. & Leonard, J. (1996). Evolutionary Software Development to Support Organisational and Business Process Change: A Case Study Account. *Journal of Information Technology, 11*, 185-198.

Lehman, M. M. (1980). Programs, life cycles, and the laws of software evolution. *Proceedings of the IEEE.*

Patel, N. V. (1999). *Developing tailorable information systems through deferred system's design*. Association for Information Systems Americas Conference on Information Systems, Milwaukee, WI.

Patel, N. V. (2001). Deferred System's Design: Situated System Requirements Gathering with Hyper-Tmodeller. *AI & Society, 15*(4), 316-343.

Robinson, H., Hall, P., et al. (1998). Postmodern Software Development. *The Computer Journal, 41*(6), 363-375.

Warboys, B. C. (1991). The practical application of process modelling: Some early reflections. *Proceedings of the 1st European Workshop on Software Process Technology*. Milan: ACIA Press.

Warboys, B. C. (1994). *Reflections on the relationship between business process re-engineering and software process modelling*. Proceedings ER '94., lecture notes in computer science, Springer Verlag.

Warboys, B. C., Kawalek, P., et al. (1999). *Business Information Systems, A process approach*. London: McGraw-Hill.