IT LEGACY SYSTEMS: ENABLING ENVIRONMENTS THAT REDUCE THE LEGACY PROBLEM: A COMPLEXITY PERSPECTIVE

Prof. Eve Mitleton-Kelly
Director
Complexity Research Programme
London School of Economics
Houghton Street
London WC2A 2AE
+44 20 7635 5553
E.Mitleton-Kelly@lse.ac.uk
&
Visiting Professor
Open University

Published in Software Evolution, edited by N. H. Madhavji, M.M. Lehman, J.F. Ramil and D. Perry, Wiley, 2004

Abstract

Information technology (IT) 'legacy' systems are often seen as a problem, particularly when they are *systems that no longer support the current business objectives or are inhibiting future developments* (for example, the creation of new financial products). Many IT legacy systems are old, but there is evidence that new systems quickly become 'legacy' in the sense that they do not fully support current and future business objectives. Because the reasons for the emergence of legacy systems are not fully understood, the same behaviour is repeated.

One such reason is the mistaken belief that legacy is merely a technical issue involving only computer software and hardware. This however is often not the case. Legacy is a socio-technical issue with the 'socio' part playing a greater role than is recognised. This chapter will use two case

studies to illustrate this assertion and to suggest ways of creating an enabling environment that may reduce the legacy problem.

Complexity theory will be used to provide some insights and three concepts will be introduced: co-evolution, feedback and social-ecosystem.

Introduction

IT Legacy systems are typically large, the cost of maintaining them is very high and they tend to constrain the business from responding fast enough to changes in business strategy, as they are not sufficiently flexible to allow significant modifications. The applications supported by the legacy systems, however, are often vital to the business and to its day-to-day operations. The IT systems cannot therefore be taken down or off line for upgrading without massive disruption and high cost. In addition upgrading legacy systems is a risky operation as multiple upgrades create a very complicated system with many interdependencies that cannot be readily identified, as documentation tends to be incomplete and those who built and later modified the system are no longer available. One of the risks is that the new system will quickly become another legacy system unable to fully support constant changes in business strategy or the frequent introduction of new products. Until the underlying reasons for the creation of legacy systems are understood the cycle will keep repeating itself.

Three key concepts will be introduced - one is ***co-evolution*** or the reciprocal influence between related entities that results in a change in two or more related entities. The term *entities* is used, as the concept can apply to (a) units of analysis in different disciplines such as species in biology or organisations in the social sciences; (b) interacting departments or groups within the same organisation; or (c) different types of related organisations such as suppliers, buyers, customers, etc.

The second concept is that of ***feedback*** as this is closely related to the co-evolutionary process and in turn to the legacy problem. A deeper understanding of the two concepts will help practitioners make better use of them. One of the insights that will be offered is that feedback is not a simple linear input-process-output mechanism that can adequately be described as positive or negative. When applied to a complex evolving system like a human organisation, feedback becomes a *non-linear, multi-loop and multi-level process.*

The third concept is that of a ***social ecosystem*** which includes all the related co-evolving entities such as businesses, governments, financial institutions, regulatory and standards bodies, customers, etc. which are able to influence each other.

The central thesis of this chapter is that *if co-evolution between the business process and IT development is enabled, then the problems associated with legacy systems will be reduced*. [Coza & Lewin 1998, Liu et al 2002, Rosenkopf & Tushman 1994, Tromp & Hoffman 2003, Van de Ven & Garud 1994] Two case studies, one with an international Bank and another with a UK Building Society, will be used to illustrate how co-evolution was facilitated and thus helped to reduce legacy problems - but if this process is to continue beyond individual projects then the organisation needs to create *enabling environments* that will facilitate the reduction of the legacy problem, by identifying both the social and the technical conditions that will help it do so. This chapter emphasises the *relationship* between the business and IT domains by focussing on the social context. Although it discusses some technical issues it does not deal with them in detail.

In the Bank case, for example, the notoriously difficult relationship between the system developers and the business users was enabled and this in turn created an environment which had a significant impact on the technical development of the system such that it ensured that the project was delivered on time, when the delivery date was critical for the introduction of the common

European currency. The Bank case study will be used to illustrate the co-evolutionary and feedback processes. These are just two principles of complex evolving systems, identified by complexity theory.

The Building Society case study will be used to introduce and illustrate some additional complexity theory principles. When these principles are understood then businesses can work with them rather than working inadvertently against them and they can be used to create enabling environments. [1]

[1] Both case studies were part of a 3-year research project led by the Complexity Group at the London School of Economics. This was one of 30 research projects funded by the UK's Engineering and Physical Science Research Council (EPSRC). In 1996 the Council set up a managed research programme entitled Systems Engineering for Business Process Change (SEBPC) with a total fund of £4.5m. The aim of the programme *was to release the full potential of IT as an enabler of business process change, and to overcome the disabling effects, which the build-up of legacy systems has on such change*. [Preface, Henderson P. 2000]

Before going any further, however, it may be useful to explain briefly what is meant by **complexity theory**. The theory explains the behaviour of systems that (a) have many interacting parts; (b) are able to adapt and co-evolve with a changing environment; and (c) are able to create 'new order' in the form of new structures, or patterns of behaviour, or properties. Complexity theory applies to the kind of complex systems that demonstrate all three characteristics. Most machine-type systems that have many interacting parts, but are for example unable to create new order, would be called 'complicated' rather than 'complex'. All human systems and consequently organisations are complex evolving systems. [Mitleton-Kelly 2003a describes ten of the principles of complex evolving systems]

The first part of the chapter will explain what is meant by the legacy problem and introduce the three concepts of *feedback, co-evolution* and *social ecosystem*; the second part will describe the two case studies; part three will discuss the socio-technical enabling environments created in the Bank and the Building Society and some insights will be summarised in the conclusion.

1. The Legacy Problem

There are several definitions of IT legacy systems [see papers in Henderson P. 2000 and 2001; see also www.dur.ac.uk/CSM/SABA/legacy-sig/; Liu et al 2002; Reddy & Reddy 2002; Tromp & Hoffman 2003] and they include the standard definition of 'legacy' as a valuable inheritance as well as the idea that the system is old and obsolete. In this context, the definition used by Tromp & Hoffman [2003] provides a good starting point and supports the findings that will be presented in this chapter.

> *"A legacy system is an operational system that has been designed, implemented and installed in a radically different environment than that imposed by the current IT strategy."*

To bring this definition in line with the argument of the chapter, the following qualification should be added *"and no longer supports the current business strategy"*.

The IT legacy problem is usually associated with old and large systems, written in assembly or an early version of a third generation language. They have been developed 20-30 years ago without anticipating that they would still be running decades later. The architectures and technology used to build the systems were relatively inflexible, and they had not been designed to accommodate such a magnitude of change over an extended period of time. The software systems have been changed extensively, but in an incremental and ad hoc manner. This provided the required improvement in functionality in the short term, but at the cost of increased connectivity and inter-dependence, and with relatively poor system understanding. Moreover, they are associated with high maintenance costs and they have become very difficult and expensive to change to further support the business objectives. When the balance between the technical and business dimension is lost, legacy can be seen as a *gap* between the business needs and the technical capabilities.

Once legacy is seen as that gap, then it is no longer confined to old systems. *New* systems may quickly become 'legacy systems' in the sense that they do not meet the full requirements of the users and are unable to fully support business evolution. This is often the outcome of a lack of understanding and communication between IT professionals and the business IT-users and strategists, or lack of adequate and appropriate feedback, which leads to separate evolutionary paths and to a divergence of interests and hence to differing future directions. In such cases there has been a low rate of co-evolution or interaction leading to reciprocal influence and change, between the business and IT domains.

For legacy problems to be reduced, the IT professionals need to understand the business process, its language, values, direction and future development, if they are going to provide IT systems which support the business. On the other hand, the business users and strategists need to understand the technical potential as well as the limits of the IT systems. Since the individuals who are attracted to the IT and business domains are psychologically and culturally different, this kind of interaction and mutual understanding is neither simple nor easy, and it certainly does not happen as a matter of course. A study carried out by Mitleton-Kelly in 85 organisations, between 1988 and 1992, interviewing over 300 business and IT strategists, indicated that communication, which leads to a deep understanding of the other domain was very rare. When it did occur it depended on specific individuals taking the initiative. The general interaction between the business and IT domains, however, was limited to occasional formal exchanges when necessary. There was little regular informal interaction and the professionals in each area of operation felt uncomfortable with the other. The two case studies showed that this was not always the case and that with the right enabling environment co-evolution and feedback between the two sets of strategists was facilitated.

1.1 Feedback

In an engineering context feedback is understood as a mechanism or a loop linking a linear input-process-output system and feedback mechanisms tend to fall into two types: (a) positive (reinforcing or amplifying) feedback and (b) negative (balancing, moderating or dampening) feedback The former is likely to create change while the latter creates stability and tends to be predictable.

Feedback in human systems however, cannot be a simple linear process with predictable and determined outputs. Actions and behaviours vary with different individuals, as well as with time and context. When applied to a complex evolving system like a human organisation, feedback becomes a *non-linear, multi-loop and multi-level process* operating at different scales - at the level of the individual, the group, the organisation, the industry, the economy, etc. [Lehman 1996, 1997, Rosenkopf & Tushman 1994, Van de Ven & Garud 1994, Bateson 1993, Doyle et al 1992] Feedback in a social system's context will be defined as *influence, which changes potential action and behaviour*.

1.2 Co-evolution

Co-evolution in both biological and social systems is taken to mean that *the evolution of one domain is partially dependent on the evolution of the other* [Erlich & Raven 1964, Kauffman 1993, 1995 (a) & (b), Koza & Lewin 1998, McKelvey 1999 (a) & (b), Pianka 1994] *or that one domain changes in the context of the other.* In human systems, co-evolution focuses on the ***relationship*** between the co-evolving entities and can be defined as the *reciprocal interactions among entities at all levels of analysis* [McKelvey 1999b] *that result in* ***reciprocal change*** (the term *entity* is used as a generic term which can apply at different scales to individuals, teams, organisations, industries, economies, etc; it can also apply to non-human artefacts such as IT systems that interact with human users, developers, etc).

There are two issues to note, one is that co-evolution takes place when related entities influence and change each other (sometimes in very subtle ways - the co-evolutionary process does not necessarily imply large or significant change; when two individuals are in discussion and each begins to see things a little differently as a result of that interaction, co-evolution has taken place). The other point to note is that co-evolution happens at all scales of interaction. It can happen between individuals and teams (e.g. software engineers, IT developers, users, business project managers, strategists, etc); between individuals and artefacts (IT systems); between departments, organisations, industries, economies, etc. The other issue is that co-evolution can only take place within an ecosystem. (This point will be explored further in 1.3.)

1.3 The Social Ecosystem

In biology an ecosystem is an environment where "each kind of organism has, as parts of its environment, other organisms of the same kind and of different kinds" that interact. [Kauffman 1993 p.242] In an organisational context, a ***social ecosystem*** is the broader social environment that contains all related businesses, within the same and other industries, suppliers, customers and shareholders, as well as the financial, economic, political and legal systems that can have an influence upon and are influenced by each other. Rosenkopf and Tushman [1994] describe the social ecosystem as an 'organisational community' and define it as *"the set of organizations that are stakeholders for a particular technology. Depending on the technology, this set of organizations can include suppliers, manufacturers, user groups, governmental agencies, standards bodies, and professional associations".* Since each entity needs to evolve in the context of other related entities, co-evolution cannot take place in isolation - it takes place within an ecosystem and it does so at all scales. Within a social context, the notion of 'ecosystem' therefore can apply both within the organisation and to the broader environment.

Hence **a complex co-evolving ecosystem is one of intricate and multiple intertwined interactions and relationships**. It isn't just a nested hierarchy of 'levels' but also of multi-directional influences and links, both direct and many-removed. Connectivity and interdependence propagate the effects of actions, decisions and behaviours throughout the ecosystem, but that propagation or influence is not uniform as it depends on the ***degree of connectedness***, which in turn affects the quality and type of feedback.

Both biological and social systems are not fully connected but display different degrees of connectedness over time. "Real (biological) ecosystems are not totally connected. Typically each species interacts with a subset of the total number of other species, hence the system has some extended web structure." [Kauffman 1993, p.255] In human ecosystems the same is true. There are networks of relationships with different degrees of connectedness. Degree of connectedness means strength of coupling and the dependencies known as ***epistatic interactions*** - i.e. the fitness contribution made by one individual will depend upon related individuals. This is a contextual measure of dependency, of direct or indirect influence that each entity has on those it is related to or is coupled with. Each individual belongs to many groups and different contexts and his/her contribution in each context depends partly on the other individuals within that group and the way they relate with the individual in question. Consider how the same individual can behave in a different way and show a range of characteristics in various contexts - part of the reason is how others within each group influence the behaviour and consequently the contribution that each individual member feels able to make. Degree of connectedness, dependency or epistatic interaction may determine the strength of feedback.

Linking the above, it is suggested that *legacy is the outcome of restricted co-evolution and inadequate feedback* between the changing business process and IT development. **The business and IT domains are evolving along two separate evolutionary paths with minimal co-evolution.**

The two domains exist within multiple environments, i.e. within business, market and technological environments, which are themselves changing. As these entities interact, they co-create their co-evolving social ecosystem. However, *weak coupling* such as infrequent interaction and/or lack of understanding and knowledge about the other's domain, lowers the *rate of co-evolution*, and creates legacy systems, which do not support the changing business process.

The Mitleton-Kelly 1988 - 1992 study showed that the relationship between IT professionals and business strategists or project managers was consistently restrained and this exacerbated isolated evolution. They only talked to each other when absolutely necessary. The users drew up requirements without understanding what the technology was able to offer; these were often minimal lists of functionality and did not explain the broader business context. Future plans for new products or a change in strategy were not communicated to the developers, as a matter of course. They therefore had to design or modify the IT system with only a very limited view of what was required. There was rarely any open, extended, face-to-face discussion of future plans and requirements and of consequent implications for system design. There was therefore very limited and highly constrained co-evolution.

To understand co-evolution, a distinction needs to be made between adaptation *to* a changing environment and co-evolution *within an ecosystem* [Mitleton-Kelly & Papaefthimiou 2000] Adaptation to, implies that there is a hard boundary between the system (or the organisation) and its environment. While co-evolution places the system within its ecosystem and assumes flexible boundaries between inter-related elements that reciprocally influence each other. For example, when suppliers become 'partners' or when end-users participate in the design of a new IT system and become part of the design team, the boundaries of identity and relationship change.

The emphasis therefore changes from a simple relationship between the system and its environment to a complex relationship between multiple interacting elements within a social ecosystem, co-evolving *with* each other. In this context feedback can be seen as those processes that influence change in decisions, actions and behaviours between the multiple differentially coupled entities. In one sense the *feedback loop* becomes a *multi-dimensional spiral* as each change in one entity may trigger a change in a related entity, which in turn may trigger other changes in its coupled entities. Just to complicate matters, this is not a linear causal process in the sense that change A *causes* change B. Many changes e.g. A,D,G,M, etc may *together* contribute to change B. *The reciprocal influences or feedback processes are neither uniform nor universal. They depend on* *the degree of connectedness, on epistatic interactions and on time and context.* Furthermore, the consequences of actions and decisions are again not totally determined or fully predictable and there is always a range of possible consequences (or possible futures) arising from each decision or action taken.

2. The Two Case Studies

One case study is with an international Bank and the other with a UK Building Society. They were part of a larger research project carried out by the author and colleagues. Data was collected through semi-structured interviews with business users, systems developers, business and IT strategists. Part of the methodology [Mitleton-Kelly 2003b] was to identify and study a *natural experiment*, in the sense that a group of individuals developed a different way of working and relating, which was different from the established working practice, and which could not be supported by the dominant culture of the organisation. (The experiment was undertaken by those involved, it was not 'done to them' by senior managers or researchers.) The dominant culture of the Bank, for example, supported a particular way of relating and working, which had inadvertently contributed to the legacy problem. A different way needed to be found and the UK office created a completely new way. Although certain individuals took particular actions, no one was deliberately *orchestrating* the process. Certain socio-technical conditions were introduced which encouraged and supported a different type of interaction and this facilitated the co-evolutionary process. In other words, certain individuals in the Bank's UK office initiated the conditions that facilitated the creation of a new enabling environment, which helped ameliorate the legacy problem.

The **Bank** is an international bank, but the case study concentrated on its European operations prior to the introduction of the common European currency, the Euro. The main European system was on two hardware bases. Eleven European countries, with smaller branches, using HP hardware, were serviced from the USA. While the larger branches, with IBM systems were run from the UK. Originally the IBM system was implemented in seven different countries and it started in the late 70s, early 80s, as a branch or country-centric system, referred to as *"a Bank in a box"* and it ran all the local Bank's operations. Since then, the Bank has gone through several phases of restructuring. The first set of changes in the mid-80s was to regionalise the environment, that is the hardware and the software were brought into central service centres and the branches were run remotely. The branch users run their terminals connected over leased lines into one of the service centres. These centres were subsequently centralised in the UK. This involved two phases: moving the technology and then the branch back-office processing.

These changes were part of a co-evolutionary process, in the sense that, the **organisational restructuring** (a social aspect) **changed the systems' architecture** (a technical aspect) when the Bank went though its various restructuring phases and the architecture in turn affected the ways of working.

The **Building Society** (BS) in the UK redesigned its IT systems to meet two objectives (a) to enable new insurance products to be designed and marketed within a shorter period of time, and (b) to migrate all the old products to the Future Product Framework (FPF) system, which would serve as a single processing engine. Regarding the first objective, it was expected that many problems would be resolved in the release process and testing, and new products would be marketed within a couple of weeks rather than 8 weeks. To achieve the second objective, FPF was to be used as a single processor, by using standard building blocks to design a new product. This would enable the organisation to react very quickly to the marketplace.

Although FPF was initially perceived as a solution to legacy, it did not fulfil that aim. On the contrary, it showed signs of becoming another legacy system. Part of the reason was that the patterns of behaviour that had created the old legacy system were being repeated. However, as part of the process of migration and upgrading several other things happened that had a beneficial effect on the relationship between the business and IT domains and these will be outlined below.

This study is used for three reasons:

(a) Unless patterns of behaviour that create legacy systems are understood and changed, the legacy problem will continue to recur; it is therefore essential to learn from these experiences.

(b) Some conditions were created that did improve the relationship between the two domains; if they are acknowledged and reinforced they will form the basis of an enabling environment.

(c) To introduce some additional principles of complexity within an organisational context.

One of the positive things that happened was when the female IT Manager in the Building Society started to pick up the telephone on a regular basis to invite her business colleagues to discuss the current and future needs of the organisation. This was a quite a revolutionary step, but it worked and the regular feedback sessions, not only facilitated the co-evolutionary process and produced better systems, but also accelerated the ***rate of co-evolution*** and increased the speed at which modifications were made.

Before discussing the enabling environments three types of factors will be described to provide some of the background - they are (a) business and market, (b) organisation and management and (c) technology. The distinction between the three factors is primarily conceptual, in the sense that it offers a framework for understanding the interdependence and interrelationship between them. It is also relevant for recognising and creating the complex socio-technical conditions, which enable co-evolution between the business and IT domains.

2.1 Business and market

In the Bank, *changes in business processes,* products and services had an *impact* on the Bank's *technological infrastructure.* For example, new business development in other geographical areas and changing business objectives often required the development of a new system or enhancements of the existing systems. Other examples include intensifying **competition** and the need to offer **new products** to respond to market forces. Offering new products demanded changes in the existing systems to accommodate new functionality or the development of a new system that had to interface with the existing ones. Further, **changing customer expectations** that demanded sophisticated service, affected the way information was provided by the current software infrastructure. This resulted in the need for building new interfaces to support the information. Furthermore, the economic climate and the market exerted financial pressures that affected the allocation of funds to build or rebuild an application. As a consequence new applications were often built on old technology or incremental functionality was added onto the existing system, which in turn contributed to the problem of legacy. Another way of looking at these influences is that co-evolution needs to take place at all levels: from the macro level between the organisation and its social ecosystem (which includes all related businesses, customers, competitors and suppliers as well as the economic and cultural environment) to various micro levels within the organisation. Furthermore, changes at the organisational macro level affected the various inter-related micro levels within the organisation, such as the IT systems.

In the Building Society, *changes in the strategic focus* of the organisation (whether for example, the priority lay with insurance products sold directly or through intermediaries or through cross-selling), had important *implications* for the *technological infrastructure*. The *lifecycle of some insurance products,* like pensions, also contributed to the legacy problem as they are very long (around 25-30 years). Even if a product is withdrawn from the market the IT application that supports it cannot be "*switched off*" for a number of years, until all existing policies have reached maturity. This partly explains the existence of many interconnected systems, of different technological characteristics and ages of systems that run in parallel. This interconnection and interdependency of systems again exacerbates the legacy problem.

Changes in legislation have an impact on the business in terms of the products sold and the systems that support these products have to be adjusted to accommodate the new regulations. These adjustments might range from simple code upgrades to changing the system itself, alterations to other systems that interface with the original system, the development of a new system that will interface with other older systems or all the above. The influence of exogenous institutional factors, like legislation, is also part of the feedback process, which impacts decisions, IT systems and ways of working and also contributes to the legacy problem.

Social co-evolution is dependent on feedback in the form of information and the communication of that information, and in turn influences and may even shape feedback channels of communication.

2.2 Organisation and management

In both case studies, some of the legacy issues were closely linked to the human and organisational context, such as the delivery of applications. Short cuts and compromises were made to the systems' capabilities and frequently only a part of the original specification was delivered. This resulted in incremental system enhancements and eventually to complicated and problematic applications. Some more specific problems were: (1) The **communication gap** between the developer and user communities further impeded the development process due to the different views and use of different languages (e.g. IT and business jargon). As a consequence of this poor appreciation of each other's domain, developers did not deliver according to users' expectations. (2)

The **lack of skills** to maintain the legacy systems was another consideration. It was difficult and expensive to recruit people who had knowledge of the old systems since current training was focused on the current rather that the older technologies. Furthermore, resistance to change at times prevented some people from moving away from the old technology. Consequently, some new applications did not benefit from the state-of-the-art technologies and the legacy problem was perpetuated. (3) The age of employees as well as exposure to new technology contributed to an individual's attitude towards change, but the attitude of the organisation was also important, as this directly affected the support of **training** and **education** in the workplace. (4) Personal **career agendas** were sometimes in conflict with underlying business needs. Younger employees were keen to use the latest technological tools to improve their CV. This attitude was reinforced by senior managers who committed more time and resources to the development of new systems while ignoring the old legacy systems. As they were often in place for only a short time (e.g. only 2 years in the Bank) they wanted to be associated with introducing "new sexy technology". There was not much kudos in being associated with the old legacy systems. *"Obsession with the new technology"* and personal choices in moving on with one's career seemed to override some of the underlying needs of the organisation, such as maintaining and upgrading the old legacy systems which were still essential to the business. (5) **Management discontinuity** further exacerbated the problem. The managers responsible for new initiatives did not as a rule stay in their job long enough to complete a project and to make any real impact and as a result projects were often not completed as each new manager wanted to introduce new ideas rather than complete those initiated by his predecessor.

2.3 Technology

Some of the issues associated with the technology, in both cases, were: (i) Rapid technological change and the need to keep up with current technology exerted a constant pressure on management, which had to be offset against the cost of the investment. (ii) The existing technological infrastructure, in combination with the increasing obsolescence of technology failed to meet emerging expectations and to keep up with new business requirements. (iii) Alignment and interfacing between existing and new technology (in terms of new platforms, new hardware, new software and processes) introduced multifarious problems contributing to institutional friction.

2.4 Interactions between the various elements

In the Bank case study, a high degree of interconnectivity and interdependence between the b*usiness, market, organisational* and *technical* elements created a complex social ecosystem which influenced and impacted both the business process and the IT systems. The co-evolutionary processes supported by feedback influences, included the following interactions, which have been simplified for ease of illustration: changes in the business and the market necessitated changes in products. This in turn meant adjustments to the existing applications. After many repetitions of this process, positive feedback created applications with cumulative incremental enhancements, which exacerbated the legacy problem. Yet each enhancement worked in the short term and created a balance between the business need and its IT support – i.e. there was short term balancing (negative) feedback. In the longer term, however, each short-term adjustment added to the legacy problem. The legacy systems in turn constrained the business from offering new products. This was *a continuous reinforcing process, interspersed with occasional balancing processes.* In other words there were **multiple feedback processes** 'embedded' within each other. *Co-evolution took place in the sense that each domain* (i.e. IT systems and business process*) changed in the context of the other, and in turn influenced each other.* Coupled interactions and feedback processes, therefore, contributed to the creation of a problem space associated with legacy systems, which constrained the way business could evolve. Yet each attempt to aid business evolution reinforced the legacy problem.

Following are some examples of how interacting elements created the legacy systems problem in the Bank case study. (For a more detailed account of this case study, please see Mitleton-Kelly & Papaefthimiou 2000).

a. One element arises from increasing interconnectivity and interdependence among the **system components and the applications**. The Bank services *"very high value global corporate clients"*. The basis of that service is that it will provide those customers with the technology infrastructure to support their business. This means that the Bank will often customise or engineer solutions into its systems, and change their coded components, to support individual clients. Over time a layered system infrastructure was created, which was tailored to service many different customers. The interconnectivity and interdependence become so intricately intertwined that a point is reached when *"to undo that complexity is almost insurmountable without going back to the business perspective and understanding where those customers are going and whether they are willing to accept a change in the way that we're working with them which allows us to undo some of the legacy and therefore some of the complexity."* (Bank Senior IT Manager) An important point to note is that emphasis was placed on the **relationship** between the business and its customers and the IT developers, that was based on a reciprocal understanding of business direction (related to the future needs of customers) and IT constraints. This kind of relationship, leading to an understanding of each other's domain, helps create the environment that enables co-evolution.

b. Another element contributing to the operational complexity of the socio-technical system was that **organisational restructuring** (a social aspect) **changed the systems' architecture** (a technical aspect) when the Bank went though its various restructuring phases in the 80s and 90s.

c. The **identification of ownership of common components** and of the need for upgrading was much more difficult as multiple owners had to be identified and to be persuaded of the benefits, before they would sign off. The technical problems associated with upgrading the systems, impacted the organisational issue of ownership and the geographically dispersed organisational structure added to the problem. The multi-ownership issue did not arise with systems that were managed and owned locally in a single country. This example shows how the intricate interrelationship and feedback processes through influence and impact, between technological and organisational factors, creates the complex problem space of legacy: a technical problem impacted an organisational issue while organisational changes exacerbated the technical concerns.

d. Another aspect was that the Bank had made a conscious effort to try and isolate modules of the legacy 'Bank in a box' system and to **create stand-alone components**, which still communicated with it. They were Windows NT based front-end servers. They had not succeeded in replacing the full set of legacy software and the partial replacements used current technology. In an effort to update the system with new technology (instead of incremental adjustment) it had created new complex interfaces with the old systems.

e. Another element contributing to the legacy problem was that the maintenance and further development of the IT systems had been centralised within the UK group, which controlled 16 systems on both HP and IBM platforms. Thus, as resources for the maintenance and support were held centrally, **local knowledge** of the branch technology of the system was lost. Hence an organisational issue (centralisation and cost reduction) affected the technology infrastructure, which in turn affected the knowledge base. This may have consequences on the future maintenance of the local systems and on local business.

The above examples illustrated the complex interactions of diverse geographic, business, organisational and technical elements, within a co-evolving social ecosystem.

3 The Socio-Technical Enabling Environment

3.1 The Bank's Enabling Environment

Despite the above and other problems, the Bank project was completed successfully. One of the main drivers was the exogenous pressure of legal and regulatory requirements imposed by the European Union, which needed to be implemented before the Bank was ready to handle the common European currency. However, although the exogenous pressure was a necessary condition, it was not sufficient for success. Many other conditions needed to be created internally and this section describes some of them, which contributed to a local *socio-technical enabling environment.*

The project introduced new technologies, and because of its high profile was also able to import an international team of technical experts. But what facilitated the technical success were certain social conditions initiated by the Project Manager in charge of the project. One of the most important aspects was the facilitation of a closer working relationship between the business and information systems professionals. The Project Manager introduced a regular monthly meeting where all developers, business project managers and operational staff met to update each other on progress. There were two rules: (a) everyone had to make time to attend the meeting and (b) they all used the minimum of jargon and tried to be as intelligible as possible to those not familiar with their specialist field. The first couple of meetings were strained and tense - but once connections and insights started happening about how problems could be resolved or how the technology could be used, or understanding what the business wanted and needed to achieve, the meetings became generative and creative and were welcomed. But what they achieved beyond a 'feel good' atmosphere was that it facilitated feedback and the co-evolutionary process between the developers and the business managers and it accelerated the rate of that co-evolution to a significant degree.

An enabling environment was created that included the following social conditions:

- New procedures introducing regular monthly meetings, which enabled *good networking* and *trust*, as well as a *common language* leading to mutual *understanding*.
- *Autonomy*: the project manager was allowed to introduce the new procedures, without interference.
- A *senior manager supported* the changes, but did not interfere with the process.
- *Stability*: sufficient *continuity* to see the project through, in an environment where constant change of personnel was a given.
- An *interpreter* mediated the dialogue between the domains. This ensured understanding on both sides but also protected the technologists from constant minor changes in requirements.

The monthly meetings, supported by weekly information updates, enabled the three environments of technology, business and operations to talk together regularly and in a way that was going against established ways of working. In time, the various stakeholders involved in the projects began to identify **cross-dependencies in terms of the business project relationships,** which led to new insights, and new ways of working. Once the conditions were provided the individuals involved were able to self-organise, to make the necessary decisions and take the appropriate actions. (*Self-organisation* occurs in a social context when a group of individuals decide to undertake a certain task that is not part of their daily routine - they decide what to do, when and how to do it, without anyone outside the group directing their actions.) This illustrates micro-agent interaction, at the level of individuals and groups, which is neither managed nor controlled from the top. Once the inhibitors were removed and the enablers put in place, new behaviours and ways of working emerged. (*Emergent* properties, patterns, qualities or behaviours are more than the sum of the parts and tend to be unpredictable. It is the non-linear, non-determined interactions between the parts or micro-agents that create emergent patterns at the next macro-level.) The monthly sessions improved communication between the different domains by improving understanding, but they also allowed for the **emergence of new ways of working**, and in the process helped the business become fitter or more competitive. The point to note is that new ways of working were not designed or determined in advance. They came into being or emerged when the relationships and interactions changed.

Another important element was the **articulation of business requirements** as an iterative process with regular face-to-face meetings. These meetings were at a senior management level with (a) a vice president who owned the product, was responsible for the profit and loss and determined the business requirements; (b) a senior and experienced business project manager who was a seasoned Banker, with a good knowledge of the Bank, and (c) a senior technology project manager who defined the IS platform(s) and the technical development of the project. This constant dialogue created a willingness to **communicate** and a level of **trust,** which were essential enablers of co-evolution. These social processes can also be seen as *feedback enabling or facilitating processes.* For example, trust facilitated better communication, which in turn enabled the building of IT systems that facilitated the evolution of the business.

What was achieved took a particular individual, supported by his senior manager, to create the conditions that enabled dialogue, understanding and a good articulation of requirements. He created the initial conditions, to improve the relationship between the domains, but he could not foresee how the process would work or whether it would work. As it happened, it did work and a substantial *network rapport* was established between the domains based on **trust**, **a common language** and **mutual understanding**. They worked well together, because the conditions were right and they were prepared to *self organise* and work in a different way. The new relationships were not designed or even intended. They happened spontaneously in the sense that their emergence was enabled but not stipulated.

The achievement however, could be a one-off. Unless the new procedures and ways of working become **embedded** in the culture of the organisation, they are likely to dissipate over time. Once the initiator is no longer in place, the danger of dissipation or reversion to the dominant mode of working will assert itself. In this case there has been some embedding and some continuity, but the process is fragile. A new set of organisational changes could destroy it. Part of the embedding is the networking rapport that has been established. But the network rapport is implicit and informal, and is therefore under threat if there are too many and too frequent changes and the Bank's culture is one of constant change in management positions. *"Every two years someone else is in the post so that there is that lack of continuity."* If the rate and degree of change is too great then the network will become invalid.

An essential aspect of creating an enabling environment is the conscious appreciation of what is happening, why it is taking place and how it can be facilitated in future. Those particular types of meeting were appropriate to that organisation at a specific time. The generic principle is that improved **communication**, which includes some face-to-face meetings, is an essential element. It builds **relationships**, **networks** and **trust** and a **deeper understanding and knowledge** of other, but closely related, parts of the business. There are organisations like Shell, where building and maintaining one's network within the organisation is recognised as an essential part of one's career plan. The Shell culture acknowledges the importance of networks and actively facilitates their construction. But this is not enough. There are other social and technical conditions that need to be acknowledged and facilitated such as **supporting training and education in new technologies**, while at the same time appreciating the importance of the deep knowledge that IS professionals develop over years working with particular systems, that are old but essential to the business. The loss of knowledge and expertise also applies to **local knowledge** (technical and business) when restructuring and centralisation take place. Once lost it is very difficult (often impossible) to resurrect and much time and effort can be wasted in re-learning what has been lost. A degree of continuity may also be necessary. Constant movement of personnel because of restructuring or specific policies, needs to be kept under review and be flexible.

3.2 The Building Society & Some Complexity Principles

The emphasis in the Bank study was on the interaction of multiple socio-technical elements at micro and macro levels of interaction, focussing on co-evolution and feedback processes. It also used the example of a natural experiment to illustrate how some enabling conditions helped create a new way of working and relating, that was different from the dominant culture. The Building Society case study will be used to introduce and illustrate some additional principles of complex evolving systems, identified by complexity theory. When these principles are understood then they can be used to create enabling environments.

a. *"Gurus" as emergent phenomena; operating far-from-equilibrium and exploration of the space of possibilities.* The part functionality and shortcomings of the legacy systems, the continuous changes and enhancements, and the difficulty involved in the process due to lack of proper documentation gave rise to the so-called "**system experts**" or "**gurus**". These people had invaluable system knowledge and expertise and had either a business or a technology background. The "experts" from the business side, acted as interpreters between the business users and the IT developers by helping to translate business requirements into technical language. This helped to overcome the communication problem between the business users and the IT developers. While the technical gurus had a deep knowledge of the undocumented legacy system and were able to help the new developers navigate its intricacies.

The "gurus" ***emerged*** out of necessity. They were not appointed and no one defined their job description. Lack of skills, lack of system knowledge, and lack of documentation, exacerbated when IT professionals moved, retired or left the company, acted as a constraint to business evolution. ***Constraints*** are not always undesirable, as they can force both the individual and the organisation to find a different way of working, which can often be innovative, to overcome the constraint. A trivial but illustrative analogy is a boulder in the middle of a stream of water. It cannot be moved, but the water can flow around it, perhaps cutting new channels in the process. The organisation therefore had to find a different way of operating by ***exploring its space of possibilities*** or possible alternatives. Exploration is not always explicit or systematic - it can be intuitive and is often quite creative, particularly when trying to overcome a constraint. It can however be restricted if there is a risk-averse or a blame culture that does not encourage experimentation. By definition, when one is trying out different alternatives a few will work and many will not. But to find the ones that work people need to work through some that do not. Exploration therefore carries the risk of failure - but if the alternatives that do not work are seen as part of the exploration process, then employees may be encouraged to try out new, creative procedures that are not the norm. The corollary of this freedom however is responsibility. In organisations where this approach has worked, each individual carried and was aware of carrying responsibility for their actions and whatever new ideas they tried out they could not risk injuring the organisation.

One way of looking at the process is that constraints may push the organisation ***far-from-equilibrium***, in the sense that they push it away from the standard way of working, away from the norm. The gurus are not the norm, there is no career path or job description for them and no one could have predicted their emergence. When pushed far-from-the-norm individuals and organisations are forced to explore alternatives. This exploration may be deliberate or it could be implicit and emergent. However, exploration needs to be enabled and emergent properties need to be recognised and not inhibited. In this case the gurus enabled a different way of working, and helped to overcome certain constraints, which could have had a deleterious effect on the development of the business.

b. Self-organised informal networks, epistatic interactions and connectedness. One multi-disciplinary project on legacy systems, in the Building Society, brought together various experts. They found that they worked well together and could help each other. This was a new departure in established ways of working. Once that project was completed the team was disbanded, but the ***informal network*** it created, has since been often resurrected, on a self-organised basis. Whenever there is a project related to IT legacy systems, people in the network call each other and try to work on the project together, on an informal basis. Because of their previous experience of working together, they know each other's expertise and can call on those with the necessary knowledge. No manager external to the group dictates or directs these interactions. The individuals within the self-organised group initiate them. This is self-organisation at a micro-scale where individuals take the initiative to talk to others and to carry out tasks they recognise as necessary. With improved communication, results were always good. The enablers here were knowledge of available skills and expertise gained through the initial project. But subsequently, flexibility in allowing self-organised groups to work together helped. However, to create a robust enabling environment, it would be necessary to acknowledge the value of such interactions and actively encourage them. Both the self-organised groups and the gurus are also illustrations of ***epistatic*** **interaction**. The contribution of each individual depended on those other individuals he/she worked with, and was enhanced in particular contexts. The quality of contribution or epistatic interaction also depended on the degree of **connectedness**. Networks or webs are not constantly connected [Kauffman 1993, 1995a]. Their robustness depends on their ability to re-establish dormant connections, when necessary. But the quality, density or intensity of the connections, even between the same individuals, varies over time; hence the degree of connectedness is not a constant.

c. Legacy as positive feedback and pattern repetition. The way management viewed the legacy systems, and continuation of the same processes reinforced the legacy systems. The business, organisation and technology processes interacted with each other on established and **repeated patterns** to produce more legacy. Once a pattern of interaction was established it continued to reinforce itself through a positive feedback process**.**

Even when the organisation has explored its space of possibilities and introduced new technology, established thinking, ways of working and relating can counteract and reduce the expected advantages. The Building Society had implemented a new approach to systems development that could reduce time to market for new insurance products, from 8 to 2 weeks. This would enable the organisation to co-evolve quickly with its marketplace. However, despite all the expectations, the mindsets, technology procedures and ways of working which originally helped create the old legacy systems, were being repeated. The repetitions of patterns of behaviour, as **reinforcing feedback processes**, recreated the legacy problem. In this case it was important to recognise what was happening and to break the cycle. This process had begun by encouraging the marketing people to build new applications, but going against well-established norms is difficult and needs constant support and acknowledgement. An important initiative had been taken, what was needed was stronger and continuous support as well as recognition of the difference the new procedures made when a new application created a new product quickly in response to market demand.

4. Summary and Conclusions

This chapter has examined the nature of IT legacy systems and explored some of the factors that created them. An important insight has been that legacy is not a purely technical issue but a socio-technical one and some of the co-evolutionary and feedback processes that contributed to legacy were described. Another key insight was that in most organisations IT development and the business process tend to interact minimally and the two domains often evolve in isolation from one another. If however, co-evolution between them were facilitated, the legacy problem might be reduced.

It was suggested that legacy arises from a multiplicity of intricately inter-related and inter-dependent socio-technical factors which influence and change each other, through multiple, but inadequate feedback processes. Feedback in complex social systems is based on multi-loop, multi-level processes, at many inter-related micro and macro levels. Emergence operates at the micro-macro interactions, but self-organisation, far-from-equilibrium conditions, and exploration of the space of possibilities are also operating at cross-entity interactions within a co-evolving social ecosystem. Reinforcing and balancing feedback mechanisms introduce change and stability respectively, and may operate sequentially or in parallel. The feedback loops, which take place at both micro-agent and macro-structure levels, vary in their intensity and influence. They may be imagined as *a plethora of interacting and interconnected micro-feedback-processes whose connectivity and inter-action creates emergent macro-feedback-processes and structures*

The chapter looked at two case studies with severe legacy problems. They each tried to resolve the problem by facilitating new ways of working and relating and by creating an environment that facilitated interaction. Both the Bank and the Building Society case studies emphasised the importance of **communication, trust** and **understanding** as essential feedback processes facilitating co-evolution between the business and IT domains.

The Bank case study was used to illustrate the interaction of diverse elements and their feedback processes, in terms of influence, within their social ecosystem. The elements chosen were the organisational, market and technological environments and their influence on business evolution and IT development. The case also showed the relationship between micro-agent interaction and macro level relationships, within a social ecosystem. Finally some of the conditions that enabled co-evolution were identified, both between the business and IT domains, and between the organisational, market and technological environments. The Building Society case study was used to illustrate (a) how repeated patterns of behaviour recreate legacy systems and (b) some complexity principles and their contribution to the creation of an enabling environment.

Enabling environments may be transient but can also be made more robust if the underlying principles of how organisations function, as complex social systems are better understood. [Mitleton-Kelly 2003a] For example, providing the conditions for generative interaction and then allowing the individuals and groups involved to work out their own way of working, often creates innovative and more efficient and effective procedures. There needs to be a balance between the prescribed and the emergent to allow space for self-organisation and a culture that encourages a degree of risk taking in the exploration of the space of possibilities; but employees need also to appreciate that they are responsible and they cannot risk the well-being of the organisation.

Most change initiatives, try to define or design the new organisational form (of a department or a whole organisation) and its interactions in detail. The logic of enabling environments on the other hand argues for providing the social and technical conditions that offer all the necessary support but then allow for emergence and self-organisation. In addition, awareness of the importance of facilitating co-evolution and how its rate can be accelerated could have a significant impact on resolving extremely difficult issues such as the IT legacy problem. Finally it is essential to appreciate that complexity theory principles do not work in isolation and their inter-relationship needs to be understood - e.g. co-evolution is dependent on feedback and so is emergence, while connectivity and interdependence are necessary to all of them.

Acknowledgements

The paper is based on research enabled by two EPSRC awards under the SEBPC Programme: IT & Computer Science Programme (GR/MO2590).

An earlier version of the paper has been published as Mitleton-Kelly & Papaefthimiou 2001. Maria-Christiana Papaefthimiou co-authored that paper and I am grateful to her for her contribution.

References

Note: An extensive bibliography on complexity is available on

http://www.lse.ac.uk/complexity

Bateson R.N. 1993 ‘Introduction to Control System Technology’ Prentice-Hall, Englewood Cliffs, NJ

Doyle J.C., Francis B.A & Tannenbaum A.R. 1992 'Feedback Control Theory' MacMillan, New York

Ehrlich P R & Raven P H, 1964 'Butterflies and plants: A study in Co-evolution' *Evolution*, no.18, 586–608

Henderson P. 2000, **Ed.** *'Systems Engineering for Business Process Change'*, Springer-Verlag 2000, ISBN-1-85233-222-0

Henderson P. 2001, Ed. *'Systems Engineering for Business Process Change'* Vol. 2, Springer Verlag 2001, ISBN 1-85233-399-5

Liu K, Sun L & Bennett K 2002 *"Co-Design of Business and IT Systems"* Introduction by Guest Editors in Information Systems Frontiers 4:3, 251-256, 2002 ABI/INFORM Global

Kauffman S, 1993 'The Origins of Order: Self-Organisation and Selection in Evolution', Oxford University Press

Kauffman S, 1995a 'At Home in the Universe', Penguin

Kauffman S & Macready W, 1995b 'Technological Evolution and Adaptive Organizations' *Complexity* vol. 1, no. 2, 26-43

Koza M P & Lewin A, 1998 'The Co-evolution of Strategic Alliances' *Organization Science*, no. 9, 255–264

Lehman M M, 1996 'Feedback in the Software Evolution Process' in Information & Software Technology 38, 681-686, Elsevier

Lehman M M, 1997 Laws of Software Evolution Revisited, position paper, *EWSPT96*, Oct. 1996, LNCS 1149, Springer Verlag, 1997, pp. 108-124,

McKelvey B, 1999a 'Self-organization, Complexity Catastrophe, and Microstate Models at the

Edge of Chaos', in Baum J A C & and McKelvey B (eds), *Variations in Organization Science: In Honor of Donald T. Campbell*, Thousand Oaks, CA, Sage, 279–307

McKelvey B, 1999b 'Visionary Leadership vs Distributed Intelligence: Strategy, Microcoevolution, Complexity' in *Proceedings of EIASM Workshop*, Brussels, June 1999

Mitleton-Kelly E, 1988 - 1992 study (unpublished) in 85 organisations. EMK interviewed over 300 business and IT strategists looking at the relationship between the two domains. The study was carried out in the UK and the USA - primarily New York and San Francisco.

Mitleton-Kelly E & Papaefthimiou M-C, 2000 *'Co-evolution and an Enabling Environment: A Solution To Legacy?*' in Henderson P (Ed) *Systems Engineering for Business Process Change*, Springer-Verlag; ISBN-1-85233-222-0

Mitleton-Kelly E. and Papaefthimiou MC. 2001 *'Co-Evolution of Diverse Elements Interacting within a Social Ecosystem'* in *'Systems Engineering for Business Process Change'* Vol. 2, Edited by Henderson P., Springer-Verlag; ISBN 1-85233-399-5

Mitleton-Kelly E. **2003a** '*Ten Principles of Complexity & Enabling Infrastructures' in* 'Complex Systems & Evolutionary Perspectives of Organisations: The Application of Complexity Theory to Organisations' Edited by Mitleton-Kelly, selected papers on complexity by 14 international authors, Elsevier 2003, ISBN 0-08-043957-8

Mitleton-Kelly E. 2003b *'Complexity Research - Approaches and Methods: The LSE Complexity Group Integrated Methodology'* in Keskinen A, Aaltonen M, Mitleton-Kelly E *"Organisational Complexity"*. Foreword by Stuart Kauffman. Scientific Papers 1/2003, TUTU Publications, Finland Futures Research Centre, Helsinki, 2003

Pianka E R, 1994 *'Evolutionary Ecology'* HarperCollins, New York

Reddy S. B. & Reddy R. 2002 '*Competitive agility and the challenge of legacy information* *systems'* Industrial Management & Data Systems, 1 March 2002, vol. 102, no. 1, pp. 5-16(12) MCB University Press

Rosenkopf L & Tushman M.L. 1994 *'The Co-Evolution of Technology and Organization'* in Joel A.C. Baum and Jitendra V Singh (Eds), Evolutionary Dynamics of Organizations, Oxford University Press

Tromp H & Hoffman G 2003 "*Evolution of legacy systems'* Elisa ICSM workshop 2003

Van de Ven A.H. & Garud R. 1994 *'The Coevolution of Technical and Institutional Events in the Development of an Innovation'* in Joel A. C. Baum and Jitendra V. Singh (Eds.), Evolutionary Dynamics of Organizations, New York/Oxford, Oxford University Press