PERGAMON International Journal of Information Management 21 (2001) 365–384

International Journal of
Information
Management

www.elsevier.com/locate/ijinfomgt

# Organisational scenarios and legacy systems

Carole Brooke[a,*], Magnus Ramage[b]

[a] *Faculty of Business and Management, University of Lincoln, Lincoln, UK*
[b] *Centre for Complexity & Change, Open University, UK*

**Abstract**

A legacy system is made up of technical components and social factors (such as software, people, skills, business processes) which no longer meet the needs of the business environment. The study of legacy systems has tended to be biased towards a software engineering perspective and to concentrate on technical properties. This paper suggests that the evaluation of potential change options for legacy systems can only be carried out as part of an holistic organisational analysis. That is, the evaluation of legacy systems must take place within a framework that combines business and technical considerations. In particular, we believe that the business strategy must lead this process. Accordingly, we have designed an inter-disciplinary approach which brings together an organisational scenarios tool (based on concepts from the field of organisational development) and a technical scenarios tool (based on concepts from the field of software engineering). These tools are applied in an iterative way, so that technical options are tested out against the business needs. It is, thus, a dynamic tool which seeks to mimic the nature of organisational change, as far as is practicable. The research project described here is entitled software as a business asset (SABA) and was funded by the Engineering and Physical Sciences Research Council (EPSRC) under the systems engineering for business process change (SEBPC) programme. This paper describes the research approach and its iterative stages, and illustrates its use within a large engineering firm (Engco). Its application produced useful insights for the organisation, as well as pointers for further modification of our research approach. 

*Keywords:* Legacy system; Scenario; Business process; Software

## 1. An inter-disciplinary approach to legacy systems

The definition of a legacy system is, in itself, currently a subject of some debate amongst academics and practitioners. During the early stages of our research project considerable time was spent comparing various definitions from the literature and from experience of working with

*Corresponding author. Tel.: +44-1522-886342; fax: +44-1522-886032.
*E-mail addresses:* cbrooke@lincoln.ac.uk (C. Brooke), m.ramage@open.ac.uk (M. Ramage).

0268-4012/01/$ - see front matter 
PII: S0268-4012(01)00023-8

computer practitioners. The result of this work to date is a paper which argues for a broader definition than has hitherto been offered by researchers (Gold, 1998). The key features of this definition are summarised here. Also, our understanding of the nature of legacy systems and their problems has been helped greatly by a series of workshops. Working with more than ninety academics and industrialists, we have pooled our collective understanding of legacy systems and shared approaches to solving the problems of such systems (Ramage & Munro, 2000).

Crucially, we recognise the need to distinguish between legacy software and legacy systems. Legacy software is critical software that cannot be modified efficiently. In other words, it is software perceived by the business to be critical to its operations, and yet difficult to modify without incurring great expense (in terms of time, skill, etc). Legacy software is often described as being any or all of the following: large, old, heavily modified, difficult to maintain, old-fashioned. Cynics reading the popular computing press might suppose that legacy software is a term invented by late capitalism, intended to make perfectly good software look outdated and thus sell more products. This view is also rather narrow. The problem is not so much the software per se, but the gap between the capability of the entire computer system and the needs of the business in which it is used. We need to look beyond the software in order to understand why legacy presents a problem for business.

In contrast, a legacy system refers to much more than the software. It is a wider system of which the software is merely a part. Other components of the system might include: people, expertise, hardware, data, business processes, and approaches to software maintenance and development. Understanding a legacy system also requires taking account of its relationship to the business environment. All these things—and especially their interactions with each other—constitute a legacy system. Thus, legacy systems consist of much more than just a technical dimension: they encompass issues of organisational structure, strategy, process, and workflow. This is a much more complex consideration.

The word 'legacy' also relates to that which is left after a particular event occurs (typically, someone's death). What, then, might this *legacy event* be in terms of legacy systems? Contemporary examples at a macro level might include the millennium 'bug' or the introduction of the Euro. At a micro level, a legacy event will vary from company to company. The key point for all of these examples is that the event doesn't exist until it is recognised by somebody. In addition, the event may not be a one-off, and it could also result from internal organisational circumstances as well as external ones.

A legacy system is one which no longer meet the needs of its organisational environment. Such systems cause considerable problems in modern organisations, from recent high-profile cases such as the millennium bug to many less widespread but equally serious occasions when mergers have been cancelled or new product launches delayed because the existing technology was unable to change sufficiently quickly. As Liu, Albert, Bernadette, Hanifa, and Alan (1998) have said:

> To remain competitive businesses must continually change their processes, sometimes radically, though more often incrementally, to cope with their changing environment. As a result, IT systems become inadequate in reflecting business needs, either operationally or economically, and so become legacy systems.

It may be inferred from what we have said so far that the term 'legacy' is entirely negative. This is not necessarily the case in practice. Many organisations have a great deal of valuable data, functionality, encoding of processes and expertise bound up in their legacy systems. Sometimes, the organisation may view the system as an organisational memory. Their objective is not necessarily to eradicate the legacy but to enable it to endure into the future. Attempts have been made by some to avoid confusion by replacing the term legacy with something else in order to transcend its negative connotations; for instance, the term heritage systems. Nevertheless, these neologisms quickly take on the same connotations and become just as problematic. Rather, we prefer to continue to use the established term of legacy. In doing so, we remind ourselves that the meaning of the word 'legacy' in everyday speech also implies something that can be handed on to future generations; something that can have a positive value.

At some point, an organisation will decide that its legacy is endangered and will wish to make changes to its systems to ensure that its value is not lost. At what point this decision is made will be entirely subjective and different for each organisation (not least depending on who makes the decision and for what particular reasons). *Legacy is in the eye of the beholder*. We realised, therefore, early on in the project that devising a rigid set of alternative solutions (such as would be the case in software re-engineering) would be an unsuitable way forward for this research. Nevertheless, it is possible to devise ways to help organisations identify the nature of the legacy problem and to assist them in identifying alternative approaches to change. We recognised a need to help organisations be proactive in identifying what systems might become legacy, to enable them to prepare for change, and to ensure that any technical changes were driven by business needs, rather than vice versa. Ultimately, we wanted to ensure that assets remained assets and that potential liabilities were minimised. In this way, our work became focused on facilitating and supporting organisations rather than on prescribing solutions. This was reflected in the stated remit of our project which was "*to develop approaches to assist business to make decisions about legacy systems*".

Adopting such a broad definition of legacy systems requires a research approach that is inter-disciplinary. Our study has developed an approach that combines technical expertise from the software engineering field with theory and method from the organisational development arena. In doing so, the research project aims to contribute at a methodological as well as a practical level. Bringing together the different knowledge bases of our respective disciplines has helped us to adopt a more holistic approach and to enrich our understanding of the organisational issues. So far, we have found very few research teams to be working in this way (this was one of the reasons why the EPSRC set up the SEBPC programme of which this project is a part). We have found the research experience to be very valuable and hope that we can encourage more research teams to work together across disciplinary divides. We also note, with interest, that we originally called our project SABA (software as a business asset). We now question the appropriateness of this in light of our research findings that have led us to view legacy systems in much broader terms than software per se. To that extent, the project title itself has become a legacy.

## 2. The SABA approach

Fig. 1 shows the SABA approach in overview.

The SABA approach is iterative, with at least one cycle being performed on each occasion. It begins with the organisational scenarios tool (OST), moves on to the technical scenarios tool

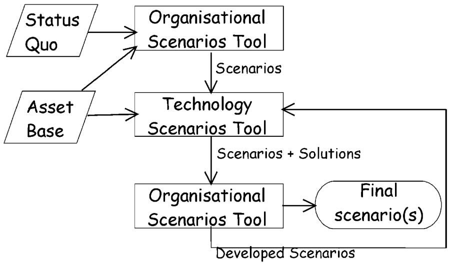


Fig. 1. Overview of the SABA model.

(TST) and then moves through the OST stage again. Participants must include (but are not restricted to) technical and business experts in order to be able to generate the information required at each stage. Ideally, a participant group should consist of about a dozen people and include:

- Senior directors (preferably including someone at Board level).
- Managers from different organisational functions (including Human Resources).
- IT specialists (preferably including a software engineer).
- Front-line staff (including those at the external customer interface).
- End users (preferably including an external customer).

All stages are heavily reliant on facilitation of the participants. This raises issues of the skill and selection of facilitators, not to mention the politics of participation. These issues are addressed towards the end of the paper.

### 2.1. The organisational scenarios tool

The initial OST stage begins with helping participants to describe their organisation as it currently exists. A useful starting point is an icebreaker exercise, such as asking participants to illustrate on paper their personal view of the legacy system. After this, the whole group together is asked to generate the first scenario, referred to as the status quo. Following this, participants are given three new scenarios to consider which are, in themselves, stereotypes but which encourage participants to be creative in thinking about organisational change. Next, participants are asked to work in small groups and develop their own scenarios. The aim is for three or four additional scenarios to be produced per group. Once completed, the status quo and the new scenarios from each group are brought together in a plenary session. Participants are asked to prioritise the scenarios in order to avoid an explosion of possibilities before moving onto the next stage. However, the status quo must always be carried forward for comparison, as the objective is to assess the gap between what is and what could be, as well as to assess practicality of suggested changes.

The OST was developed by Brooke (2000) and inspired by the research of Clegg et al. (1996). The core of the method is a matrix which is presented in Table 1. This shows that each scenario is analysed against nine separate criteria. Since the OST must begin with a description of the status quo, the first column of the matrix refers to the status quo analysed according to the nine criteria.

Table 1
The organisational scenarios tool

| | Scenario 1: status quo | Scenario 2: automate | Scenario 3: informate | Scenario 4: transform | … | Scenario *n* |
|---|---|---|---|---|---|---|
| Boundary | | | | | | |
| Vision | | | | | | |
| Logic | | | | | | |
| Structure | | | | | | |
| Roles | | | | | | |
| View of information | | | | | | |
| Costs | | | | | | |
| Benefits | | | | | | |
| Risks | | | | | | |

The next three columns refer to the three stereotypical scenarios which are given to the stakeholder groups to work with as a springboard to developing further scenarios.

The nine criteria are briefly listed here and are expanded in Table 2 (detail of the conceptual development of the nine criteria is given in Brooke, 2000). Table 2 shows the status quo of *Tiger plc*, a company in the food sector (identity protected). A further, fully worked example of the OST stage will be given of *Engco*, in the case material section later in this paper.

- Boundary: the unit of analysis (e.g. the whole organisation or one area).
- Vision: the overall business approach (e.g. specialised sales).
- Logic: organisational rationale for the vision.
- Structure: of the organisation.
- Roles: organisational roles of people.
- View of information: information as an objectified unit of resource (the resource view) versus information as a subjectively interpreted phenomenon (the perceptual view).
- Costs: major costs, both financial and non-financial.
- Benefits: both financial and non-financial.
- Risks: major sources of risk.

After the status quo scenario has been evaluated against these criteria, the three stereotypical scenarios are given to the stakeholders to consider. These are presented in Table 3. These stereotypes have been developed by Brooke (2000) from work such as Zuboff (1988) and Cash, Eccles, Nohria and Nolan (1994) and are labelled ‘automate’, ‘informate’, and ‘transform’. In order to contextualise Table 3, a brief overview only will be repeated here.

Cash et al. (1994) distinguished between the three stereotypes thus:

Automate: IT *substitutes* for human effort.
Informate: IT *augments* human effort.
Transform: IT *restructures* a set of tasks or processes.

An automate approach to IT is based on principles of the substitution of human labour and cost efficiency. The organisation’s vision will, therefore, tend to be limited to a view of the business

Table 2
The status quo at Tiger plc plotted against the nine criteria

| Criterion | Organisational features | Explanation of criterion |
|---|---|---|
| Boundary | The sales function. | The boundary refers to the scope of the current analysis—is it the whole company that is being reviewed or one particular aspect of it? |
| Vision | Specialized sales. | The vision describes the approach the company has adopted. In this example, it has chosen to treat sales as a specialised function. The rationale for this is largely because the core business is regarded as production. |
| Logic | Centralized production. | Production has been centralised and other functions are regarded as existing to support it. |
| Structure | Differentiated functions. Support production. | The structure that flows from the above arrangement is described here. The dominant feature of this company is a structure along discrete functional divides. |
| Roles | Single skilled and little variety. | The work roles associated with the functional structure of the company are clearly definable. They are fairly typically Tayloristic, narrowly skilled, specialised and consisting of little variety. |
| View of information | Tangible/measurable resources are emphasised (for example time, money, head count, equipment) rather than less quantifiable information assets (e.g. intellective and interpretative capacity). Technology tends to be defined in terms of functional areas of the business and mirrors its physical activities. | A resource view of information (rather than a perceptual view) is taken here. This reflects a rational view of organisation. A rational view gives little emphasis to the importance of the individual and their subjectivity and instead focuses on quantifiable objects. In extreme cases, information passed around the organisation (especially through information technology) will be viewed as objective fact with little or no role for individual interpretation. |
| Costs | Less variety. Few sales incentives. Labour turnover. Expensive production and sales functions. | Jobs are fairly mundane and involve little variety. Investment is mainly in the core production processes and to some extent in sales functioning. No real |

Table 2 (*continued*)

| Criterion | Organisational features | Explanation of criterion |
|---|---|---|
| | | incentive for sales staff as they are far removed from both the production process and the consumer. |
| Benefits | Simplified operation. Increased capacity. | Economies of scale are gained from this arrangement. Fairly low wage costs due to lowish level of skills employed within functions. Avoidance of complex organisational structures, so lower coordination costs. |
| Risks | Under-utilisation of skills. Poor motivation. Functional silos. | Individuals' abilities not enhanced as jobs are narrowly defined. Motivation is low and there is little communication or involvement between functions in the organisation. No incentive for staff to take a more holistic view of |

that maximises technological control, streamlines work processes, and reduces variability. Physical resources will be stressed: labour, machinery, finances, and so on. Benefits will tend to be short term, focused on reductions in overhead, labour, and increased management control. Similarly, automation does not recognise a primary role for perceptions.

The first conceptualisation of the term informate is generally credited to Zuboff (1988). An informate approach to IT is based on the principles of augmenting human effort and helping individuals to add value through the application of intellective skills. The rise of cross-functional activities, the ability to see relationships between different parts of the business, and to have more of an overview of the work context will enable individual workers to identify and anticipate new business opportunities and novel ways of working. Job expansion, quicker response times, and improved market sensitivity are some of the characteristics that define the informate scenario. The vision of the business has moved beyond specialisation and tightened control to an individualised approach that pushes more of the decision and control mechanisms out from the centre and closer to the customer interface.

Transform takes the informate scenario to its extreme. Transformation became an established term following the work of the MIT'90 s research team (Scott Morton 1991). The basic argument was that it was very tempting to apply IT in order to achieve efficiency through automation (and sometimes it was also the simpler route) but that long term benefits could only come from applying it in a way that literally transformed the business. The structure of the organisation has become much more diffuse and difficult to physically identify. Staffing and groupings are constantly changing to reflect the requirements of the business. Indeed, the very nature and purpose of the business itself requires re-evaluation. Management's role has become one of co-ordination rather than direction and control. The business needs to be very proactive in its approach to the marketplace and, therefore, innovations in structure and roles, as well as in products and services come to the fore.

Table 3
The Stereotypes: Automate, Informate, Transform

| | Automate profile | Informate profile | Transform profile |
|---|---|---|---|
| Boundary | Production department | Production department and customer-facing departments. | Whole organisation. |
| Vision | Specialised manufacturing. | Individualised products. | Innovative products. |
| Logic | Substitute human effort. Maximise time/effort/£. | Augment human effort. Release new potential. Be more responsive. | Restructure. Become more proactive. |
| Structure | Specialised functions. Centralised. | Cross-functional activity and use of networking. Focus on intra-organisational links: 'tight–loose'. | Fully networked organisation. Groups formed as necessary. Focus on inter-organisational links: 'loose–tight'. |
| Roles | Defined by technology. Streamlined. Tendency towards specialisation. | Expanded jobs/skills developed. | Flexible/multi-skilled. Increase in contract staff. |
| View of information | Tangible/measurable resources e.g. time, money, headcount, equipment. What counts is what can be measured. | Intellective/analytical skills emphasised. Individuals shape activities and influence new direction; bring alternative interpretations. | Hybrid staff and specialists are key. Intra- and inter-organisational knowledge are highly valued. Emphasis is on co-ordination rather than direction (automate) or individualism (informate). |
| Costs | Hi-tech machinery. Capital intensive. Labour turnover. | Loss of control. Co-ordination. Decision risks. | Depends on restructure but could be: re-location, re-design of systems, re-skilling of staff and management |
| Benefits | Cheaper labour. Predictable output. Greater control. | Quicker response times. Job satisfaction. Market sensitivity. | Responsive to changing context. Develop new alliances. Global positioning. |
| Risks | De-humanises workforce. 'Leaves the brains at the door'. Driven by the technology. | Misses potential. Failure to re-evaluate the core business areas. | Exploitation of workforce. No economy of scale. Sense of insecurity. Loss of planning and continuity. |

So far as Table 3 is concerned, the transformate scenario represents a deeper structural change than the informate scenario. Whereas informating might involve the restructuring of jobs and roles, full transformation involves a re-evaluation of the very nature and purpose of the business itself.

These stereotypical scenarios encourage participants to think in progressively more radical ways about the organisation. It is also important that they identify both the positive and negative effects of transformation, informating and automation, and articulate their views on these. Participants are then encouraged to use the same nine criteria to generate their own organisational scenarios (usually between three or four of their own).

By the end of the OST phase, participants have identified several possible organisational scenarios, each with different implications for technology design and use. These must then be prioritised. Once the participants have decided which of the new scenarios are to be carried forward to the next stage, the facilitator helps them construct one or more technological portfolios. The portfolio is a *technical list* that describes the technology (IT) needed to support each scenario. Importantly, this is done by the same participants as before, and is not dominated by technologists. Once this has been done, the participants are ready to move onto the next stage.

## 2.2. The technical scenarios tool

The objective of this phase of the model is to identify, for each of the scenarios, the possibilities for technology change, resulting in a set of technological options. In particular, this will require analysis of existing IT systems that might have been loosely classified as legacy. An overview of the tool can be seen in Fig. 2, although it is important to note that a waterfall process model is only used for illustration; in practice there will be considerable iteration between the stages of the tool.

The first stage involves *information capture*; the asset base discovered during the OST is analysed to add the more detailed information needed for software engineering methods. This includes details of the software itself (such as languages, structure, change history, documentation, data organisation) as well as details of the software maintenance processes, the staff undertaking these, domain information, and the role of the software in the organisation. We are basing our approach on established techniques such as (Swanson & Beath, 1989; Vergugo, 1988). In our experience, it is possible to home in quickly on the underlying problem (e.g. data usage) and to avoid wasting time on irrelevant matters. It is important to note that this stage is mostly passive: information will be provided to the analysis stage as various questions in that stage require it.

In conjunction with the above, potential *solution routes* will be determined (different information will be required for different solutions). The archetype solutions are those identified such as discard, reverse engineer etc., recalling that different solutions may apply to different parts

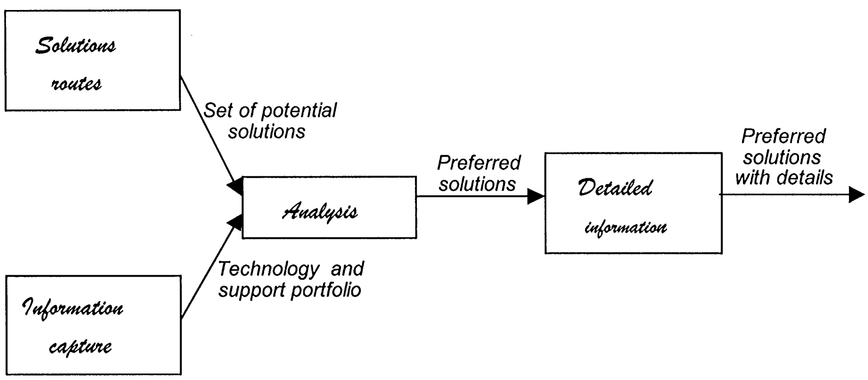


Fig. 2. The TST in overview.

of the software. Again, it seems possible to prune out some solutions quickly which are manifestly unsuitable for meeting the scenario objectives. For example, software reengineering, with its conservative approach to preserving business processes, will not work if the processes of the organisation are being changed. Support for this stage can mostly be handled by a simple questionnaire-type approach, using questions for each such as:

- *Leave*: Has a strategic decision already been made to change the system?
- *Discard*: Is the system part of the core processes? Would the company cease to exist in its current form without it?
- '*Re-require*' *(update existing requirements and build a new system)*: What is the magnitude of task? What requirements do we have already? When were they written?
- *Reengineering*: Is source code available? Do tools exist to analyse the code, or do they need to be built? What language is the code written in? Was any development method used (e.g. Jackson)? Has the method been followed consistently in past maintenance? How big is the system (lines of code)?

The results of these first two stages feed into the *analysis* stage. This is the point at which participants are assisted to come to a decision for each scenario about which methods are most appropriate for their organisation. This is basically a matter of weighing the capabilities of the different methods against the nature of the technical requirements of the organisation. When this kind of analysis has been conducted in general by consultants on software maintenance, it has tended to be by expert opinion: with their knowledge of the available solutions and the situation, they have reached a judgement about appropriate methods to solve the legacy problems. However, our goal in this project is to develop approaches that will allow members of organisations who have much less expert knowledge of this area to make such decisions. Time is also an issue here: it could take months to conduct a full analysis of the appropriateness of just one of the potential methods (e.g., encapsulation); this time is clearly not going to be available in a lot of cases, and it may be possible to quickly close down a large number of solution routes.

Once a set of preferred solutions has been identified, it will be essential to seek more detailed information. This will include costs, risks, tool investment, staffing and expertise, as well as more detailed technical information about the software itself. Our approach is based on the work of Foster (1993), Sneed (1995) and also the Renaissance project (1998). Also linked to this will be a process of evaluation, which seeks to re-examine the business impact of the solutions found in the TST. This will draw on various methods from information systems evaluation, and evaluation methods which combine technical and organisational perspectives, such as constructive evaluation (Stenning, 1999) and systemic evaluation for stakeholder learning (Ramage, 1998).

### 2.3. The second iteration

At this point, we have a set of organisational scenarios (from the OST) and a set of preferred technical solutions for each of those scenarios (from the TST). The next step is to identify the preferred solution that will best meet the needs of the preferred scenario. We see this being undertaken by expert software engineers. In practice, whichever solution is adopted will be a compromise involving engineering trade-offs. It is essential, therefore, that the results of the TST stage are evaluated for their business impact, either through the evaluation phase mentioned above or, preferrably, through a second iteration of the OST.

A second iteration of the model is preferable because a key objective of the SABA approach is to help an organisation assess its future ability to change. Passing the results of the TST through the OST stage again will illustrate an organisation's capacity for change within the context of the first change. This attempt at 'future proofing' will make a significant contribution to the evaluation of organisational change, particularly at strategic levels.

### *2.4. Engco—a real life case example*

The case example presented here concerns '*Engco*' (true identity protected for reasons of confidentiality). Engco is a medium-sized engineering company, part of a large European group. They have a manufacturing resource planning (MRP/MRPII) system, to handle order design and fulfillment, stock control and cost monitoring. For the purposes of this paper we will call this system COMA. COMA runs on a mainframe that is now less commonly used. The underlying software was once popular but the company is now the largest user in the world, and it is likely that the suppliers will end their support in the medium to long term. This is clearly a critical risk to the company; however, they recently examined the main enterprise resource planning systems (SAP, Baan etc.) as possible alternatives, and concluded that none met their needs sufficiently well. So, for the moment, they have decided to stay with their existing system. They acknowledged that they were, thus, in a classical late-maintenance phase with the vendor and wished to identify ways to address the situation.

We conducted two one-day workshops working through the SABA approach with members of the organisation. Given constraints on the timing of the research, we ran with six participants rather than twelve. However, key stakeholders were represented in the group—senior management, technical experts and end users—and so we were satisfied that the principle of a broad range of participation was met. It was important for our development of the SABA approach, however, to note that some flexibility in participation is inevitably called for when dealing with business environments.

Originally, we had planned to work through a full iteration of the SABA model but because of sudden and substantial changes in Engco's structural set-up between the first and second workshops, Engco preferred to focus on the OST stages of the approach. Consequently, we spent less time on detail at the TST stage, although we did elicit some critical points and began to sketch out possible solution routes with them. It was tempting at the time to assume that the reality of business life was preventing us from fully exercising our research approach and that we might have to re-design SABA to take account of these experiences. With the benefit of hindsight, however, we realised that this was not the case. We discovered that the real value of the approach was not so much in the outputs at each stage but in the processes that went to make them up. Even if only the OST stage is considered, by the end of the workshops, Engco had gained some valuable insights into their need for change.

Before moving on to consider these insights, we need to discuss the workshops in more detail.

We conducted the workshops in a fairly informal way. At the beginning of the first day, the participants agreed that the boundary of analysis for the exercise should be the whole supply chain (from parts to delivery) which the COMA system supports. The participants were then asked to produce diagrams or pictures illustrating their personal view of the COMA system. This exercise served as an icebreaker and immediately highlighted some key features of COMA. Three examples serve here to indicate some of the inherent problems with the system. COMA was represented as:

1. A broken railway line: with COMA as a large train running away in one direction, with a gap between it and the customers at either end (one set sitting at a station, another pushing hard at an old manual train).
2. A safe with two doors into it (i.e. not as many as there might be): materials and stock are going in, and demands, information and products are coming out.
3. At the centre of everything: the system produces lots of paper, receiving timely deliveries but containing too much stock and almost, but not quite, connected to lots of remote sites.

Having discussed these representations and established some of the issues to be addressed, the participants went on to produce the status quo scenario. This was done in a plenary session led by one of the SABA research team. Once this was completed, the three stereotypes were explained (automate, informate, transform) and then the group were asked to generate more scenarios of their own. A member of the research team was available to help them with their analysis if needed. However, little intervention was required as the participants found the OST technique fairly easy to apply. After a few hours they presented three new scenarios. These are presented in Table 4 and can be summarised as follows:

Scenario 5: A focus on new product development—increase market share by having something new to offer. Always keep ahead of the competition's product range. Be proactive.

Scenario 6: Make customer support the company's main aim; look to nurture existing customers, and build a long-term, ongoing relationship with them.

Scenario 7: A hybrid of the first two, which would make customisation and design the key feature of the company. This scenario views the company as a general provider of products which produce heat and power, rather than being tied to a specific technology.

The hybrid scenario had been favoured at the first workshop but because of the major changes in between times, by the second workshop the customer support option was prioritised. To quote one participant: "*because we haven't done [scenario 5] very well*". The explanation they gave for this change of heart was that with hindsight they felt their track record of new product development was poor. The exercise also revealed a considerable perceptual gap between the head offices and the local bases. There was a corporate drive to exchange skills and information between staff, using the same technical tools if possible. The head office view was that: "*Changing a software tool should be as easy as changing a pencil*". Both the technical experts and the end users at the workshops said this was unrealistic.

Within the context of the major changes that were taking place, the revertion to the more familiar territory represented by Scenario 6 was entirely understandable, especially since policy on technology change seemed to be coming increasingly from the new parent company. Nevertheless, the scenarios produced were now a documented part of the decision-making process. The fact that the organisation could return to the various options at any time in the future to re-evaluate their position lent strength to their decision processes and made it more traceable and transparent.

It was interesting to note that no-one mentioned the word 'legacy' until about 15 min before the end of the first workshop. This seemed a very positive sign despite the problems raised during the

Table 4
Scenarios generated by Engco

| | Scenario 1: status quo | Scenario 5: new product development | Scenario 6: support existing products | Scenario 7: hybrid |
|---|---|---|---|---|
| Boundary | Product supply chain | Increasing market share. | Consolidation of customer base with existing product base, with limited new developments. | Customer-centric innovation. |
| Vision | Customer satisfaction; quality product. | Greater sales; longevity; corporate image. | Worldwide support organisation; deepen rather than expand customer relationships. | Provision of power and heat. |
| Logic | Product driven | Changing environment—customer demand changes (e.g. fuel types). Flexibility of product. | Customer partnership; different regions can focus on well-understood markets. | Power is always in demand. Fuel based innovation within an appropriately efficient manner. |
| Structure | Hierarchical/functional. Do whatever works. | Cross-functional teams—engineering and manufacturing specialists and external members. | Distributed, decentralised generalist; with support from centralised specialists. Based on customer types. | Specialist design and manufacture teams supported by specialist. Customer relationship managers. |
| Roles | Some specialised, some multi-functional. Trying to evolve. | Communication, team working, specialists. | Flexible, multi-skilled, locally employed. | Distributed intelligence, local control/ co-ordination.Temporary technical specialists vs. permanent managers. |
| View of information | Short-term, keen on predictable information. Look at Now. Largely concerned with tangible information. | Predictive modelling—no surprises expected. | Centrally controlled, well distributed. Locally focused view; generalised view held centrally. | Matrix-like combination of technical requirements and customer management. |

(*continued on next page*)

Table 4 (*continued*)

| | Scenario 1: status quo | Scenario 5: new product development | Scenario 6: support existing products | Scenario 7: hybrid |
|---|---|---|---|---|
| Costs | Alternative to COMA would cost lots of money (organisation doesn't currently take IT people costs into account). | Manufacturing—on edge; damage to corporate reputation. | Distribution, multi-site transport/distribution/ purchasing. | Information—provision, management. Customer relationship managers and company-wide training. |
| Benefits | Secure data, easy to manage, well understood. Not seen as a problem, non-threatening. Powerful user group. | Fast introduction of new products. Higher business profile. Family concept. | Scale; breadth of market provides comfort. | Customer satisfaction; increased market share; decreased market loss. |
| Risks | User base shrinking; vendor may eventually decide not to support COMA. Little room for change. Difficult relationship with general IT strategy of parent company. | Not what the customer wants. Development overhead. | Market death; fragmentation of manufacturing styles. | Loss of commercially sensitive facilities. Loss of control. |

icebreaker exercise. It was clear that the status quo scenario generated, while perhaps not as flexible as it might be, was just as valid as the other scenarios generated during the workshop. Overall, the feeling was that for the future the customer support scenario, building on existing strengths, was preferred, with the status quo as a fallback option.

We then went on to explore the technical scenarios tool. For the reasons mentioned earlier, less emphasis was given by Engco to this stage than to the OST, although many issues were surfaced which were important for identifying preferred solution routes.

In accordance with the information capture stage of the TST, the research team elicited technical information from the participants. Discussion included the current software portfolio, where the software supported, where it constrained, what flexibility it allowed, implications for the future, and which people were involved with it. The team then talked through the possible solution routes with the whole group. What follows is a summary of the results of this analysis stage.

### *2.5. Leave*

Engco cannot ignore the need for change since the vendor has confirmed that support for COMA is in question even for the medium term. At one extreme, Engco could buy in a service to sort out its data, and let the outsource company take all the technology risk. They did not favour this option, though, feeling that they would lose too much control. An alternative was to purchase the source code from the vendor. However, that presented problems in itself, given that it is written in a ‘quirky’ fourth generation language. It was thought that some parts of COMA could remain, and some form of wrapping could take place. Wrapping would consist of boxing off parts of the system (e.g. the data dictionary) so that it would be more self-contained. Since COMA is basically modular in form, this should be possible. It was felt that in areas outside COMA the corporate offering was much weaker. This suggested Engco had scope for setting standards themselves.

### *2.6. Discard*

The system could be completely replaced by another choice (e.g. SAP or Bahn). One possibility was that the replacement could be delivered directly to them. Engco would then cease to do any maintenance themselves. Again, there was a feeling that too much control might be lost to outsiders with this solution.

### *2.7. Re-require (update existing requirements and build a new system)*

It would be possible to upgrade the existing system using a ‘screen-scraping, smoke and mirrors’ approach. This meant the users would get the impression they were using different systems. Purchasing would see a completely different COMA to manufacturing, for instance. In fact, it would be the same system but with user interfaces (screens) tailored to their particular activities. As the group said:

> The users won’t even know what’s happening underneath.

### 2.8. Re-engineer

It seemed that the preferred solution route was to re-engineer the data contained within COMA and re-build the system to meet its new requirements. However, Engco said they had attempted too many initiatives at once:

> We always want to do 100% of everything but we can only do 80%, so we do nothing.

On reflection they felt they could only really do small things in parallel and check that they were done correctly, ensuring that regular contact was maintained between the end users and the technical experts. This was attempted through 'technology demos' where potential users were shown what technology is available. However, they were careful to stress that leading edge technology (ironically termed 'bleeding edge') was not generally useful in itself. In some circumstances it had value, but their significant concern for the human issues was evident.

During the OST stage, the organisation had evaluated itself as biased towards a resource view of information (see Table 4). Yet this did not adequately reflect their prioritised scenarios which were based on visions of flexible, customer-focused strategies. As they put it:

> Data is not seen as being owned by users, but as stewarded by them.

This quote, in particular, highlights some interesting contradictions—the tension between the more quantifiable aspects of systems development (a resource-driven view) and a more people-focused (perceptual) view; and a tension between a desire for flexibility and a need for internal power and control. They summed this up by saying:

> Put good people in the right place, support them well and trust them. Sell them a vision and take the implementation as read.

When asked what would deliver a radical, ten-fold improvement in value for the organisation their response was *data*. The concept of software as a business asset was less important at Engco than *information* being seen as a business assset (see our earlier comment about the SABA title as a legacy in itself). They have a constant need to update data from one file format to another, as the old one becomes obsolete. Deciding to upgrade thousands of engineering drawings to a new format is a big decision, and making sure a format is selected which will be useful in the future is difficult. The question raised was not so much about re-engineering code as re-engineering data. They argued that the data was longer-lived and of strategic importance to the company, especially when it became information, and then knowledge.

Although Engco regarded data as their main asset, they confessed they were not getting to grips with issues of knowledge management. They recognised that making a business case for long-term knowledge management projects was crucial, but they were finding it very problematic. Benefits were seen as being gained in the short term. Long-term projects were still seen in terms of a cost rather than a clear benefit and, as one Director said:

> the main decision-making criterion is profit: does a change in IT make/save money for the company?

It seemed a resource view of information was affecting their ability to tackle the more perceptual issues of knowledge management, yet they realised this ability was critical to the organisational strategy. This also echoed a statement made by Engco's Director of Information Technology at a meeting with the research team earlier in the year. He had commented that the company were

> …getting better at developing our view of information—we see the difference now between data, information and knowledge.

The SABA approach had shown that Engco knew where they needed to be but were struggling to implement the changes in attitude and values that were required.

## 3. Conclusions

The subjectivity of the SABA approach is both a strength and a challenge. In particular, the OST is dependent for its exact form on the workshop design and facilitation style of those using it. It is designed to be a tool-in-use, modifiable according to the context. It is not a tool that can be picked up and used simplistically in the same way by everyone. We regard this context dependency to be methodologically important and this means our approach will not present the same results on all occasions. This in itself is a strength. After all, the purpose of the exercise is to assist organisations in preparing for dynamic change. At any particular moment in time, the nature of, and preference for, organisational scenarios will change (as we witnessed between the first and second Engco workshops). The scenario outcomes are expected to be dynamic and changing. It is, thus, not the specific content of each scenario that the SABA approach emphasises, but the processes it offers up to support the analysis.

Various case studies have led us to consider using SABA in situations where decisions about organisational strategy have already been taken and the questions asked are purely technical. Is SABA inappropriate for these, given the close ties between organisational and technical change in the model? Or can the TST 'component' of the model be used on its own? We may be able to persuade organisations that it is useful to reconsider their strategic choices, but we are then at risk of academic arrogance. On the other hand, there are many cases where taking a purely technical perspective (and refusing to take a wider view) can lead to exactly the legacy problems which SABA is intended to counter.

The OST is already fairly well developed, but it will be of interest to us—and we hope of particular use to organisational analysis and information systems communities—to see its use in wider practice, and to scrutinise in more detail what does or does not work for companies. For instance, the question of how to apply weighting criteria in order to prioritise one organisational scenario over another needs further consideration and will draw on research into profiling and portfolio analysis from both the management field (e.g. Kaplan & Norton, 1993; Ward, Griffiths, & Whitmore, 1992) and software maintenance (e.g. Vergugo, 1988).

Significant changes also need to be made to the technical scenarios tool to incorporate an evaluation step. Regarding the *information capture* stage, we are looking at the possible relevance of information audit techniques (e.g. see Buchanan & Gibb, 1998) to assist in the process. In addition, during the *analysis* stage there needs to be a way to assist closure around the potentially large number of solution routes that could be identified. One way of implementing this would be with a simple tool to propose suitable methods based on the characteristics of the technical requirements. This could take the form of an expert system that guides the user through the different solution options. The system would be very clearly a decision support instrument: it is not expected to reach decisions for those using it, rather to assist them in making their own decisions. In creating this tool, we would be drawing on a body of published work on the analysis of potential methods for legacy systems evolution, such as Sneed (1995), Ramage and Bennett (1998), and an unpublished report by the UK's Central Computer and Telecommunications Agency. We anticipate that only a few solutions (perhaps up to three or four) would be left at the end of this analysis process.

We envisage that the expert system would use the following stages:

(a) Take as inputs the potential solution set, basic information about the systems portfolio, and some notion of weighting of different factors.
(b) For each potential solution:
  (i) Use the basic portfolio information;
  (ii) Ask the analyst for more detailed information as appropriate;
  (iii) Calculate a single number showing the fit of this solution to the portfolio (according to the weightings given).
(c) Order the potential solutions according to the metric in (b) (ii) above, and output the list of solutions in their order, for prioritisation and then further development in the next stage (the detailed information stage).

This derives from the need for close interweaving of the business implications of the scenarios and the solution routes produced. We are especially looking at Stenning's work on constructive evaluation (Stenning, 1999) and Ramage's work on systemic evaluation for stakeholder learning (Ramage, 1998). A common theme of both these methodologies is that they can be used throughout the software lifecycle. Such an approach will help to reduce linearity in the systems evaluation process; something to which the Technical Scenarios Tool is especially prone (see Fig. 2). To expect to follow clearly defined linear steps in legacy system evaluation is as much of a fantasy as the use of the waterfall approach in software development. In practice, there must be flexibility around the components of the various stages as well as iteration between the organisational and technical parts of the analysis. To prevent the former denies the contextually dependent nature of legacy systems. To neglect the latter promotes technological determinism (i.e. letting technology drive the business need rather than vice versa).

A major contribution of the SABA project to date has been the integration of several very divergent approaches to the analysis and re-development of computer systems. Information systems, organisational analysis and software engineering presently have almost no communication between them either in academic or commercial practice and we are (with others) trying to build a bridge across this gap. Whilst our research continues and there are issues still to be addressed, the research has demonstrated that an inter-disciplinary approach to methodological development can broaden our understanding of the issues, and generate some valuable insights for organisations and researchers alike.

## Acknowledgements

The SABA project was funded by the Engineering and Physical Sciences Research Council, under their Systems Engineering for Business Process Change programme. Thanks to Engco and the workshop participants. Also to the other three members of the SABA team, Professor Keith Bennett, Professor Malcolm Munro and Nic Gold. Finally, a commentary by Vincenzo Dispenza on an early draft of this paper was extremely useful. More details of the full research project can be found at http://www.dur.ac.uk/CSM/SABA/

## References

Brooke, C. (2000). A Framework for evaluating organisational choice and process redesign issues. *Journal of Information Technology*, *15*(1), 17–28.

Cash, J. I., Eccles, R. G., Nohria, N., & Nolan, R. L. (1994). *Building the information-age organisation: Structure, control and information technologies*. Boston: Richard D. Irwin.

Clegg, C., Coleman, P., Hornby, P., Maclaren, R., Robson, J., Carey, N., & Symon, G. (1996). Tools to incorporate some psychological and organisational issues during the development of computer-based systems. *Ergonomics*, *39*(3), 482–511.

Foster, J. R. (1993). *Cost factors in software maintenance*. Ph.D. thesis, Computer Science Department, University of Durham, 1993.

Gold, N. (1998). *The meaning of 'Legacy Systems'*. Technical Report 7/98, Dept. of Computer Science, Durham University, UK. http://www.dur.ac.uk/CSM/SABA/legacy-wksp1/meaning.html.

Kaplan, R. S., & Norton, D. P. (1993). Putting the balanced scorecard to work. *Harvard Business Review, 71*(5), 134–147.

Liu, K., Albert A., Bernadette S., Hanifa S., & Alan D. (1998). Using semiotic techniques to derive requirements from legacy systems. Paper Presented at the First SEBPC Legacy Workshop. http://www.dur.ac.uk/CSM/SABA/legacy-wksp1/staffs.html.

Ramage, M. (1998). *The learning way: Evaluating co-operative systems*, Ph.D. thesis, University of Lancaster, UK.

Ramage, M., & Bennett, K. (1998). Maintaining maintainability. In Khoshgoftaar, T. M., & Bennett, K. (Eds.), *Proceedings of the IEEE International Conference on Software Maintenance*, Washington, DC, USA (pp. 275–281). California, USA: IEEE Computer Society.

Ramage, M., & Munro, M. (2000). It's not just about old software: a wider view of legacy systems. In P. Henderson (Ed.), *Systems Engineering for Business Process Change*. London: Springer.

Renaissance project (1998). *The renaissance method handbook*. http://www.comp.lancs.ac.uk/computing/research/cseg/projects/renaissance/RenaissanceWeb/project/Documents3.html (1998).

Scott Morton, M. S. (Ed.) 1991. *The Corporation of the 1990s: Information Technology and Organisational Transformation*. Oxford, UK: Oxford University Press.

Sneed, H. (1995). Planning the reengineering of legacy systems. *IEEE Software*, *12*(1), 24–34.

Stenning, V. (1999). Judgement day for IT systems. *Computer Bulletin,* Series V, *1*(2), 16–17.

Swanson, E. B., & Beath, C. M., 1989. *Maintaining information systems in organizations*. John Wiley: Chichester.

Vergugo, G. (1988). Portfolio analysis: Managing software as an asset. *Proceedings of the Sixth International Conference on Software Maintenance/Management,* Washington, DC, USA (pp. 17–24). New York, USA: US DPMA Education Foundation.

Ward, J., Griffiths, P., & Whitmore, P. (Eds.) (1992). *Strategic planning for information systems,* Chichester: Wiley Information Systems series.

Zuboff, S. (1988). *In the age of the smart machine: The future of work and power*. Oxford: Heinemann.

**Dr. Carole Brooke** is currently Reader in the Faculty of Business and Management at the University of Lincoln, UK. She was previously a Lecturer in Information Technology at Durham University Business School for five years. Her

background is inter-disciplinary including a Ph.D. in Information Technology and Business Administration from City University Business School and an MA in Archaeology and Anthropology from the University of Cambridge. She has ten years of commercial experience including insurance, Research and Development, and recruitment consultancy. Work to date has focused on the human issues of organisational change, especially relating to information technology. Amongst Dr. Brooke's current research projects are included research into the experiences of call centre employees (using critical management theories) and an exploration of community project methodologies as a way of improving stakeholder involvement in organisational change.

**Dr. Magnus Ramage** is currently a Lecturer in Information Systems at the Centre for Complexity and Change in the Open University, UK. He studied computer science and gained his Ph.D. from Lancaster University, going on to work at the University of Durham for three years as Research Assistant. During this time he worked on the EPSRC project described here, and has published a number of papers with Dr. Brooke and other members of the research team. He has also published work on scenario methods and is interested in the application of group facilitation techniques such as *open space*.