THE AUTHENTICITY OF THE RECOVERED ORIGINAL USER-REQUIREMENTS IN SYSTEMS RE-ENGINEERING

Kadida R.S. Mashaushi[*]

Abstract: The contributions of systems-re-engineering in dealing with legacy systems have been proved to be invaluable. On the other hand, the determination of original user requirements for which a legacy system was originally planned to address is such an important step in the re-engineering process that errors may invite yet another legacy system right at birth. This paper looks at how organizational-based approaches to the recovery process may produce original user requirements that may not be genuine.

BACKGROUND

It is widely acknowledged that the legacy systems phenomenon is a serious challenge to IT systems. A great deal of research work has been directed towards the discovery of effective and efficient solutions to enable these systems to continue to make valuable supporting contributions to core business operations. Since business operations are very dynamic, out of date IT systems can result in missed business opportunities and even business failure. Studies also indicate that it is not practically nor economically feasible simply to replace all existing systems (Liu *et al*, 1999). The best option to date is the re-engineering of legacy systems in order to enable them to be continually useful.

A legacy status is conferred on a software system when it cannot support new business processes. This is common because organizations tend to introduce changes to their structures and operations in order to adapt to new business environments. The importance of recovering the requirements of in-service systems has been widely acknowledged (see Liu *et al* 1999, Paul *et al*, 1997).

As has been pointed out (Liu *et al*, 1999), understanding the original user requirements which legacy systems were built to solve is the first essential and important step for any re-engineering activities. It is important and necessary to do so in order to ensure that its (software) evolution or replacement is properly informed by an understanding of what is redundant, what must be retained and what can be reused (Paul, 1997).

Understanding those user requirements is difficult for many reasons including:

- Loss of systems documentation;
- Inaccurate systems documentation;
- Absence of the original business for which requirements were based; and
- The built-in natural process of reduction of user-requirements into apparently summarized system requirements.

GENUINE ORIGINAL USER REQUIREMENT

Original and genuine user requirements can be described as 'that which the user had originally requested to be provided by the system so that the organizational goals can be achieved.' They are expressed in a way that is unambiguous, clearly telling what is to be done by the system and how it is to be done, indicating the inputs, outputs and the processes. Genuine user requirements need not indicate the technology that will be used to fulfil them. They are simply logical requirements.

McMenamin and Palmer (1984) had worked on the concepts of 'Perfect Technology' and 'the essence of a system' in relation to

[*] Lecturer, IFM

systems development, particularly in clarifying exactly what the user wants. They define the essence of a system as the activities and memory that would have to be in the system even if the system were implemented using perfect technology. In this idealistic approach, the modeler and the user are relieved from the complexities of technology-related issues that overshadow the real requirements. The following are key aspects of a system:

a) A Perfect Processor

A perfect processor would be able to do anything and everything instantly. That is, it would have infinite capabilities and infinite workload capacity. It would cost nothing, consume no energy, take no space, generate no heat, never make a mistake and never break down.

b) The Perfect Container

With the perfect container the essential systems modeling considers that there is a storage facility which can accommodate any size of storage needs. It also assumes that the facility has zero access time and cost.

So the user concentrates on his/her requirements and is given the impression that everything is possible. The resulting model represents genuine user requirements. The constraints on the requirements are introduced at a later stage during development. This approach provides a comprehensive documentation of genuine original user requirements that are free from technical, financial, economic or operational considerations.

However, most re-engineering works make extensive use of the physical incarnation to obtain original user-requirements. Approaches that have been and are being studied make use of existing elements in the software, including technology, to discover the original requirements.

These existing elements include:

- Systems documentation;
- Systems operation procedures;
- Work flows; and
- Business rules, etc.

What Might Go Wrong With Organizational Studies as Regards Genuine Original Requirements?

Some approaches to the discovery of original requirements like the AMBOLS (Liu, 1999) assume the absence of a system's documents that can aid the recovery process. Such approaches employ organizational-oriented theories as their theoretical bases in recovering the requirements. AMBOLS makes use of organizational semiotics as its theoretical foundation for the recovery processes.

In a different approach called REVERE (Paul), although documentation analysis is at the heart of the recovery process, again organizational studies also make a contribution to the recovery process.

Approaches employing organizational aspects may suffer from the following possible problems in respect of the authenticity of the requirements.

1. The systems development process and subsequent maintenance produces a final implementation design that can deviate from the user-requirements. This is because the logical model is rarely mapped fully on to an implementation model. The subsequent implementation model usually considers constraints like available technology and the procedural and operational constraints. These considerations result in inevitable deviations from the logical model (which expresses genuine user-requirements without considering how they will be implemented). It is this final implementation that is actually observed at the time of recovering the original requirements. Several real requirements might have been suppressed by the time, budget, operational or other constraints. The researcher would not be

able to see these hidden (suppressed) elements of the requirements by studying the behaviour of the existing system alone (e.g., in the absence of documents describing the essence of the original user-requirement).

2. The second problem derives from the concept that there is a relationship between action and structure. This relationship highlights how the actions of the agents of an organization are always, to some extent, modeled by the situation in which they later find themselves, that is, it leads to the recognition of the situated nature of an action (Suchman, 1983). The concept of the "situated action" has been developed and elaborated mainly by Suchman (1983. 1987) in the context concerning the problem of the relationship between the procedural specifications of an action and the action effectively carried out by an organizational agent.

   The central assertion in Suchman's work is that the "plans" (a term used by Suchman to refer to standard operating procedures, checklists, flowcharts, etc.) are not sufficient if one is really to be able to understand the action; the reality of the action can only be perceived through consideration of physical and social circumstances within which it takes place, that is, by focusing on the local and contingent interactions between the one who acts and the context. Hence the action is "situated" because it is inseparable from the place in which it unfolds. The reverse is also true.

   In the context of software and other broader sense-information systems, the actions that exhibit the behaviour of the system are in fact situational. In other words the observed actions embody in them not only the written procedures but also the environment. This means that the same procedures may produce different actions given different environments. Thus it becomes difficult to work back to the essence of a system (original user requirements) by observing its behaviour (the actions). This "situation" concept implies the presence of another form of distortion in the process of recovering the original requirements where organization analysis is employed.

3. Finally it is noted in Liu *at el* (1999) that a systems development process is reductive in nature. The different types of requirements, functional, non-functional, business and discipline, get obscured somewhere in the process of development. It is argued in Liu (1999) that the functionality provided in the design may not have a one-to-one mapping to functional user-requirements. That is to say, the design functionality may bring together different user requirements in one design element. In other words a certain number of user-requirements gets bundled into a single design element. Subsequently these design elements are translated into codes and other physical and soft forms of the final implementation.

CONCLUSION

While organizational approaches can provide useful insights into the original requirements, such an approach makes extensive use of the surrounding implementation technology. From the screen design, the workflow, file systems, reports and other visible and non-visible working elements of the systems, they represent solutions to the requirements after considering the constraints (time, budget, technology, etc.). This means that they may not represent the genuine original requirements of the user. It is in this sense that we conceive distortions in deriving the original requirements from implementation-based approaches.

Perhaps it is sensible to incorporate appropriate considerations when determining original user requirements as a first step during re-engineering of legacy systems.

REFERENCES

Ian F. Alexander (1997)
"A Historical Perspective on Requirements:" *Requirenautics Quarterly*, Issue 12, pp. 13-20.

James Power (1995)
"Institutional Approaches to Programming Language Specification," Ph.D. Dissertation.

Kecheng Liu, Albert Alderson and Zubeir Qureshi,
"Requirements Recovery From Legacy Systems by Analysing and Modelling Behaviour:" In *Proceedings of the International Conference on Software Maintenance*, IEEE Computer Society, Los Alamitos, pp. 3-12.

Kecheng Liu, Ronald K. Stamper and Kaiyin Huang (1997)
"A Morphology for re-engineering Competitive Business:" In A. Booth (Ed.), *Legacy to Client/Server - Have You Chosen Wisely/* UNICOM Press.

Palmer, J. and Macmenamin, S. (1984)
Essential Systems Analysis: John Wiley.

Paul Rayson, Roger Garside and Peter Sawyer (1999)
"Recovering Legacy Requirements:" Technical Report of the Cooperative Systems Engineering Group, Computing Department, Lancaster University.

Stefano Biazzo, (2000)
"Approaches to Business Process Analysis:" *Business Process Management Journal*, Vol. 06 Issue 2 Date 2000, ISSN 1463-7154.

Suchman, L.A. (1983)
Plans and Situated Actions: The Problem of Human-Machine Communication, Cambridge University Press, Cambridge.