CAPTURING INFORMATION SYSTEM'S REQUIREMENT USING BUSINESS PROCESS SIMULATION

Alan Serrano
Department of Information Systems and Computing
Brunel University
Uxbridge Middlesex
UB8 3PH, London, UK

KEYWORDS

Information Systems Requirements, Non-Functional Requirements, Discrete-event Simulation.

ABSTRACT

Systems requirements can be divided into two major groups: functional requirements and non-functional requirements (NFR). Functional requirements describe what the system does or is expected to do. In other words, they describe the *functionality* of the system. Non-functional requirements are concerned with describing how well the system delivers the functional requirements. Despite the fact NFR play a very important role during the software development process these have been overlooked by researchers and are less understood than other factors in software development. One reason for this might be because NFR are difficult to represent in a measurable way, something that impedes their proper analysis. This paper explores the feasibility of using Business Process Simulation (BPS) models to provide IS analyst with alternative ways to capture non-functional requirements.

INTRODUCTION

One aspect that is consistently addressed by IS development methodologies is to match organisational needs with the proposed IS. Despite this fact, however, there are still some issues that need to be overcome when capturing IS requirements. Traditional IS methodologies, such as the SDLC, reflected a view that user requirements did not change over time, and thus conceptual models could be used to represent the functionality of the system (Rolland and Prakash, 2000). This belief, however, has now being challenged. Information systems have to adapt to this environment, which in turn implies that requirements are not stable. In order to adapt to this environment, IS practitioners advocate that requirement validation must now be compared against organisational needs and not against the system functionality (Rolland and Prakash, 2000). Only in this way, they argue, can information systems adapt to the ever-changing organisational needs. To address these problems, requirements engineering extends the remit of the traditional IS modelling approach that answers the question What does the system do? to an approach that answers the question Why is the system like this?

Capturing requirements is part of the software development process and is concerned with understanding the needs and wishes of the current and new system and finding mechanisms to portray these needs. Most of the IS methodologies argue that this task should be undertaken thoroughly otherwise it could cause user dissatisfaction. Requirements Engineering (RE) is the IS domain that studies how to develop systems that meet user requirements in the best possible way. Zave and Jackson (1997) define requirements engineering as the branch of software engineering that is concerned with the analysis and capture of organisational goals considering the constraints they impose on software systems. Systems requirements can also be divided into two major groups: functional requirements and non-functional requirements (Sommerville, 1997; Bennett et al., 1999). Functional requirements describe what the system does or is expected to do. In other words, they describe the *functionality* of the system. Functional requirements representations usually illustrate the way the system operates, details of the inputs and outputs of the system, and the relationships between the data that the system will hold. Non-functional requirements are concerned with describing how well the system delivers the functional requirements. Non-functional requirements are usually expressed as performance criteria, volumes of data that the system should hold, and security considerations.

The following section describes the advantages and limitations of IS modelling techniques to capture functional and non-functional requirements.

MODELLING FUNCTIONAL REQUIREMENTS

Rich Pictures, Conceptual models, Data Flow Diagrams (DFD), Entity Relationship Diagrams (ERD), State-Transition Diagrams, IDEF1x, and Object-Orientation (OO), such as the Unified Modelling Language (UML), can be mentioned as the most dominant IS modelling techniques (Giaglis, 2001). Models in the IS domain can be used to represent many different aspects of the IS process. Consequently, the major problem when representing user requirements (functional or non-functional) is to identify which is the most appropriate

Proceedings 15th European Simulation Symposium
Alexander Verbraeck, Vlatka Hlupic (Eds.)

ISBN 3-936150-28-1 (book) / 3-936150-29-X (CD)

technique for the purposes the analyst wants to communicate.

Most of the IS modelling techniques specialise in different aspects, depending on the stage at which they are applied. For example they can be used to understand either the overall function of the system in question, to understand IS data structures, or to model the processes involved in the IS. Table 1 shows a classification of modelling techniques according to the stage that they can be applied and the aspect they address.

Table 1 Classification of Modelling Techniques (adapted from Avison and Fitzgerald, 2003)

Stage/Aspects addressed	Overall	Data	Process
Strategy	Rich Pictures		
Investigation &Analysis	Rich Pictures Objects Martices Strcuture diagrams Use Cases	Entity Modelling Class Diagrams	Data Flow Diagrams Entity Life Cycle Decision Trees Decision Tables Action Diagrams Root Definitions Conceptual Models (UML)
Logical design	Objects Matrices Structure diagrams	Normalisation Entity Modelling Class Diagrams	Decision Trees Decision Tables Action Diagrams
Implementation	Objects Matrices Structure diagrams	Normalisation	Decision Trees Decision Tables Action Diagrams

Most of the techniques described in Table 1, though, are aimed to depict functional requirements. The following section describes current alternative techniques to model non-functional requirements (NFR).

MODELLING NFR

Information systems can be determined by their functionality and also by properties of the whole system such as operational costs, performance, reliability, maintainability, portability, and many others. These constraints, also named goals, quality attributes, and *Non-functional Requirements (NFR),* play a very important role during the software development process, since they usually work as the selection criteria among a variety of decisions in the development process. Despite these facts, non-functional requirements have been overlooked by researchers and are less understood than other factors in software development. One reason for this might be because NFR are difficult to represent in a measurable way, something that impedes their proper analysis. Mylopoulos et al.(1992), Nixon (1998), and Nuseibeh and Easterbrook (2000) have identified other major problems encountered when dealing with NFR:

1. There is not a formal definition or a complete list of NFR.
2. NFR usually interact with each other, a situation that can cause conflicts and tradeoffs with implementation techniques.
3. NFR are difficult to understand and represent since they have a global impact on the future system.
4. In order to produce a system that meets the NFR, it is important to consider the organisation's characteristics. These, however, vary from one organisation to another.
5. In general, NFR represent properties of the whole system, therefore, it is almost impossible to verify them in terms of individual components.

Trying to address these problems, academics and practitioners have proposed many ways to model non-functional requirements. Mylopoulos et al., (1992) and Chung and Nixon (1995), for example, proposed a NFR framework to capture and relate non-functional requirements. The NFR framework uses a goal-oriented approach to capture NFR. After NFR are captured and analysed, the NFR framework finds links between them in order to determine the impact that a given decision would have on the requirements. Nixon (2000) extends this work and applies this framework to specify a particular group of NFR: *performance requirements*. Nixon adapts the NFR framework to integrate and catalogue a different number of knowledge of performance and information systems, including performance concepts, software performance engineering, and information systems development knowledge such as requirements, design, implementation and performance. Similar to this work, Cysneiros and do Prado Leite (1999) integrate non-functional requirements into traditional conceptual data models, namely Entity Relationships (ER). The objective is to represent NFR and understand their impact on database modelling design.

These approaches can be useful to elicit NFR and to match them against IS design in order to meet NFR. The approaches, however, do not provide the means to assess whether the proposed IS meets the NFR identified previously, neither are they useful to investigate how these requirements may affect the organisational performance.

The following sections describe the way Business Process Simulation (BPS) models can be used as a complementary technique to capture non-functional aspects of a proposed IS solution. To this end, the following section describes a case study that will be used an example to provide evidence to support this theory and subsequent sections analyse the results of the BPS models and explains the way these results can be used to capture NFR.

THE CASE STUDY

The case study presented here consists of two collaborating organisations in Greece. One company is a branch of a major multinational pharmaceuticals organisation (we will refer to this company as Org-A), while the other is a small-sized regional distributor of Org-A’s products (we will refer to this company as ‘Org-B’). The case study was carried out within a single business unit, which deals with hospital consumables. The business unit imports products from other Org-A production sites across Europe. The goods are stored in a warehouse that operates as a central despatch point for all products, which are then distributed to the company's customers via a collaborating distributor, namely Org-B. Org-B responsibilities include:

a) Maintaining an adequate inventory of products to fulfil the orders.
b) Distributing the ordered products to customer premises.

Org-B has to operate within rigorous deadlines. The agreement between the companies, stipulates that each order has to be fulfilled within 24 hours for products delivered within the city of Thessaloniki, or within 48 hours for the rest of northern Greece. Org-A management noted, however, that these targets are rarely met in practice. A brief analysis by the companies seemed to attribute the problems to some inefficiencies within the ordering system as well as difficulties being experienced by Org-B in maintaining their inventory at an optimal level. The effects that these inefficiencies caused were seen as a major source of customer dissatisfaction, so an in-depth analysis of the problem was commissioned. The main objectives of this study were:

a) To examine the existing business processes that were felt to be responsible for long lead times for order fulfilment.
b) To determine the sources of problems and propose alternative solutions.
c) To evaluate the potential of introducing appropriate IS to improve communication between the two companies.

USING BPS MODELS TO CAPTURE NFR

This section describes the way BPS models can be used to capture NFR. To this end the simulation exercise was divided in two phases: One, the development of the as-is model and two, the development of different business process scenarios (to-be models). Because this paper aims to illustrate the use of simulation models to capture NFR, the design of the BPS models will be skipped, concentrating on the results provide by such models. For more information related to design of simulation models refer to Law and Kelton (2000).

The findings of the as-is business process confirmed the concerns of the companies that delivery times were much longer than the agreed targets. Even when no backorders were required, deliveries to Thessaloniki took 38 hours (target time was 24 hours), while deliveries to the rest of northern Greece took 60 hours (target time was 48 hours). Furthermore, when those orders that had items that were out of stock were included, the average time to deliver backorders rose to 82 hours. These figures suggested that the backorders were causing severe problems, so warranted further analysis.

Any order that required some out-of-stock products would effectively result in the order being divided into two separate orders; those products that were available, and those that were out-of-stock. The available products would be delivered as soon as possible, but the out-of stock products would need to be ordered from Org-A, who would then add them to the next scheduled warehouse replenishment delivery, resulting in long delays from the order being submitted and the particular products being delivered. Hence, times were recorded for backorders. When this figure was analysed it was found that the time taken from the backorder being generated to delivery accounted for 168 hours for Thessaloniki and 190 hours for northern Greece. Consequently, a series of business process scenarios, including one using an IS solution, were designed trying to address the problems found in the as-is model. The following sub-section explains the experiments proposed.

Business Process Experimentation: To-be models

The results rendered by the as-is model indicated that order and backorder processes had some limitations in terms of process design. Thus, before proposing an IS solution, it was decided to investigate different scenarios to improve these processes without using information technology. This would help to provide a better understanding of the way processes operate and to propose an IS solution that better fits the problems found in the business process model.

Three scenarios that would possibly alleviate the problems with backorders were considered. These are described in the following list.

1. Faxing backorders. Backorders were generated by Org-B and then held until the Friday evening, before being sent by post, which takes 2 days, to Org-A. For the purposes of analysis, the solution proposed was to fax the backorders to Org-A, instead of sending them by post. It was assumed that by reducing the time the backorders spent in the mail system would have a significant impact on the delivery times.
2. Org-B sends the backorder list twice a week. Instead of waiting until Friday to send the backorder list, this is sent Tuesday and Friday evenings.
3. A final scenario that included the use of an IS system was designed. The two companies share the same database to allow Org-A to have up-to-date information on stock levels at Org-B, and hence adjust replenishment shipments accordingly. The backorder list is generated automatically so Org-A knows at any moment the real-time stock levels in Org-B warehouse. It was thought that this would have an enormous impact on the delivery times, as a backorder that was generated on a Monday could now be transmitted immediately, rather than being delayed until the Friday before being forwarded.

To-Be Business Process Results

The results obtained from running scenario 1 showed that although the time taken to send the back orders by mail was reduced from 2 days to a matter of minutes, the reduction in time taken to deliver the entire order was reduced by only 1 hour for orders and remained almost the same for backorders (times recorded for Thessaloniki). After a thorough analysis, it was found that this situation was due to two major reasons. Firstly, because of the organisation's policy, orders were retained until Friday afternoon to be faxed. Despite the fact orders arrived in a matter of minutes to Org-A, Org-A employees take about 8 hours to process an order list. Therefore, orders send by fax on Friday afternoon did not have enough time to be processed on the same day and they have to wait until Monday. In the original scenario, orders took two days to arrive at its destination by normal mail. Considering that the mail works on Saturdays, the backorder list also arrived on Monday. Hence, both scenarios process the list almost at the same time. The second, and most important reason, is that there is only one employee working in the warehouse of Org-B, who, is busy nearly 97% of his working time. Consequently, an extra experiment was performed adding one more warehouse employee, resulting in an 11 hours decrease for orders to Thessaloniki. Backorders, however, remained the same.

The second scenario was to schedule the replenishment shipments to be sent twice a week instead of once. This resulted in a reduction in delivery times for backorders, but it was much smaller than anticipated (11 hours for back orders). This was due to the same problem identified in scenario 1 related to the warehouse employee. When the time was measured combining the scenarios of having two employees, faxing backorders twice a week, ordering times were recorded as 27 (11 hours reduction) and backorders as nearly 128 (40 hours reduction).

Scenario 3 addressed the only real IS-based solution, in which, both companies share a database. Sharing a database gives Org-A a better idea of the replenishment requirements of Org-B. The results did not show a noticeable reduction in the delivery times for the orders that had in-stock products, on the contrary, an increase of 29 hours was noticed for orders. The problem, as in the previous scenarios, was due to warehouse employee workload. It was reported that he was busy 99% of his working time. The increase of utilisation (more than 2%) was due to the fact he had to deal with a slightly higher number of backorders. Those products that required backorders, however, showed a substantial reduction of nearly 74 hours. This was mainly because the backorder list would no longer need to be created, as it would be generated in conjunction with the normal replenishment shipment. A final experiment was created which combined the results from scenario 3 with an extra warehouse employee. The results were as expected, since there was a reduction of 10 hours for orders and 82 for backorders. The times reached in this scenario were the best in comparison to the as-is scenario, though, they were still distant from Org-A and Org-B targets.

The results for the average delivery times of the entire order, and the average delivery time of backorders for each of the scenarios described before are shown in Figure 1.

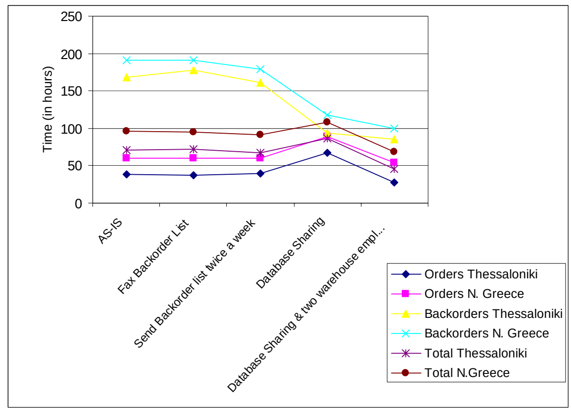


Figure 1 Delivery Times Scenarios

CONCLUSIONS

The results obtained from the business process scenarios indicated that the majority of problems of Org-A and Org-B were related to business process design and not to information technology deficiencies. It was found that a major limitation of the way Org-B operates was related to the resources assigned to the different activities performed by Org-B. It was found that the only employee working in the warehouse of Org-B was busy 97% of his working time and that changes to the business process were constrained by this issue. Because this problem is related to the way processes operate, the insertion of an IS does not provide any improvement in this respect, and the contrary may be true for some cases. It was observed that the warehouse employee needed to work more because the information system produced more workload in the activities in which this resource is used (backorders). When another warehouse employee was used in the model, many problems related to performance in the activities he was involved in were alleviated.

In order to avoid these problems, the experience gained during the BPS exercise suggests that a thorough analysis of the BP is needed, and especially, it is necessary to identify:

a) Business process limitations.
b) Different business process alternatives.

c) System bottlenecks where IS-based solutions can be implemented.

In order to identify these three issues, performance measurements of the BP are needed. To this end, the experiments proved that BPS is able to provide this information.

Perhaps, the most important outcome of the experimentation with the to-be business process models is the identification of a system bottleneck, which in turn, is the key to depict the impact of the IS on the business processes. Many business process methodologies suggest that applying IT with only the intention of automating business processes, and presumably improving performance, usually render disappointing results (Hammer, 1990; Kettinger and Groover, 1995). The results of the experiments performed on the IS-based scenario described, corroborate this statement. It was demonstrated that despite the fact that the use of technology was introduced (database sharing) the results obtained from the models were far below the expectations of the organisations. The experiments performed, however, helped to identify that the backorder sub-process was a major bottleneck of the ordering process. Initial assumptions were that the insertion of an IS would reduce the backordering processing time. Nevertheless, the results showed that this was not the case. Despite the fact that transmission times were substantially reduced, the performance of the system was still not satisfactory. To understand the reasons why the business process simulation exercise did not reflect what the IS should supposedly have delivered, the following paragraphs analyse the IS functionality in more detail.

The IS was thought to function in the following way. It was assumed that information between both companies would now be shared. Therefore, the system in Org-A would now be able to know when the stock level of any given product was below the organisation's policy. When a given stock level is below its set figures to fulfil demand, the system would set an alarm to Org-A so they could be aware that a replenishment cargo was needed. Disregarding some exceptional cases (i.e. unexpected demand in a single product) it was expected that this new way of monitoring product stock levels would reduce the number of backorders since Org-A would have information about which products needed to be replenished before they were out of stock. The business process model, however, did not represent this kind of behaviour. The business process model only reflects the time reduction due to the insertion of the IS. To reproduce the effects that this new information system would have on the business processes it is necessary to know, rather than the reduction time, the number of backorders that would be reduced due to the use of the new IS. Bearing in mind that the information system would more effectively control the inventory replenishment policies, which in turn would reduce the number of backorders produced on the system, a final experiment was proposed. The experiment consisted of gradually decreasing the percentage of backorders produced in the business processes model from 30% to 5%. The results proved that the overall ordering processing time could be reduced significantly if the percentage of backorders is reduced. The experiments showed an average of 31 hours for the delivering of orders and backorders to Thessaloniki, and 50 hours to northern Greece. These figures are closer to the targets set by both organisations.

Summarising, the simulation exercise presented in this paper provided evidence that business process simulation can be used by IS practitioners to better understand the way the organisation operates and to identify systems requirements that could be overlooked when using traditional IS modelling techniques. For example, the BPS models used in the case study helped analyst to identify that a system that could help to improve Org-A and Org-B orders delivery time should be one that not only improves processing time, but more importantly, one that reduces the number of backorders produced by the system.

FURTHER RESEARCH

The to-be BPS model provided information that can lead to the identification and measurements of NFR. For example, it was detected that to predict the impact that the introduction of an IS may have on the business processes, it was necessary to identify the number of backorders that would be produced by the new information system. These figures can only be obtained by modelling the behaviour of the information system as it is used over time. Thus, in order to portray the benefits that the use of an information system brings to the business processes it is necessary to verify that the IS satisfy certain requirements, in particular non-functional requirements.

One reason non-functional requirements are difficult to analyse is due to the fact they have a global impact on the organisation and the system itself. Similarly, it is argued that to produce systems that meet non-functional requirements it is important to consider the organisation's characteristics.

This paper used BPS techniques to represent the behaviour of the IT considering the organisational context (business process) in which they operate. One of the problems using traditional BPS techniques is that it cannot be used to portray behavioural measurements of the way a given IS functionality may provide to the BP. These measurements, according to the analysis of the results derived from this exercise are likely to be IS non-functional requirements. This paper can be used as the basis to propose an approach to depict the way IS requirements may affect BP performance with the aid of simulation techniques. Using this results and experience, a simulation framework to depict IS and BP interactions can be proposed. This framework can help to:

a) Identify non-functional requirements. Non functional requirements are not always clearly defined by the users. For example, a non-functional requirement of the case study was to offer control of the inventory system in a more efficient manner. Further analysis using the BPS model indicated that the "real" requirement was to have as few backorders as possible. This was identified by experimenting with the model and isolating the parts of the system (processes) that were concerned with the backordering process.
b) Verify that the IS satisfies non-functional requirements. Running a simulation model that depicts both IS and BP performance can be used to verify that the system satisfies user requirements.
c) Identify the variables that affect IS and/or BP performance. Once a given non-functional requirement is identified, simulation can be used to isolate the process (together with the IS used to support this process) needed to satisfy this requirement. Experimentation can be used to investigate the variables (e.g. resources, entities, etc.) that may affect IS and/or BP performance.
d) Identify conflicts and tradeoffs between non-functional requirements. Experimentation can be used to identify the effects that changes in one user requirement may have on the others.

REFERENCES

Bennett, S., McRobb, S. and Farmer, R. (1999). *Object-oriented Systems Analysis and Design Using UML*. London: McGraw-Hill.

Chung, L. and Nixon, B. A. (1995) Dealing with Non-functional Requirements: Three Experimental Studies of a Process-oriented Approach. *Proceedings of the 17th International Conference on Software Engineering*, Seattle, WA, April 24-28. pp. 25-37.

Cysneiros, L. M. and do Prado Leite, J. C. S. (1999) Integrating Non-Functional Requirements into Data Modelling. *Proceedings of the IEEE International Symposium on Requirements Engineering*, Limerick, Ireland, June 7–11. IEEE, pp. 162-171.

Giaglis, G. M. (2001) A Taxonomy of Business Process Modelling and Information Systems Modelling Techniques. *International Journal of Flexible Manufacturing Systems*, 13(2), pp. 209-228.

Hammer, M. (1990) Reengineering Work: Don't Automate, Obliterate. *Harvard Business Review*, 68(4), pp. 104-112.

Kettinger, W. J. and Groover, V. (1995) Toward a Theory of Business Process Change. *Journal of Management Information Systems*, 12(1), pp. 9-30.

Law, A. M. and Kelton, D. W. (2000). *Simulation Modelling and Analysis*. 3rd ed. New York, NY: McGraw-Hill.

Mylopoulos, J., Chung, L. and Nixon, B. A. (1992) Representing and Using Non-Functional Requirements: A Process Oriented Approach. *IEEE Transactions on Software Engineering; Special Issue on Knowledge Representation and Reasoning in Software Development*, 18(6), pp. 483-497.

Nixon, B. A. (1998) Managing Performance Requirements for Information Systems. *Proceedings of the First International Workshop on Software and Performance*, Santa Fe, New Mexico, October 12-16. ACM Press, pp. 191-144.

Nixon, B. A. (2000) Managing Performance Requirements for Information Systems. *IEEE Transactions on Software Engineering*, 26(12), pp. 1122-1146.

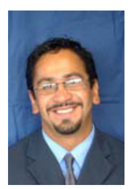
Nuseibeh, B. and Easterbrook, S. (2000) Requirements Engineering. *Proceedings of the 22th International Conference on Software Engineering*, Limerick, Ireland, June 4-11. ACM Press, pp. 35-46.

Rolland, C. and Prakash, N. (2000) From Conceptual Modelling to Requirements Engineering. *Annals of Software Engineering*, 10(1/4), pp. 151-176.

Sommerville, I. (1997). *Software Engineering*. 5th ed. Wokingham: Addison-Wesley.

Zave, P. and Jackson, M. (1997) Classification of Research Efforts in Requiremts Engineering. *ACM Computing Surveys*, 29(4), pp. 315-321.

BIOGRAPHY



ALAN SERRANO is a Lecturer of IS at Brunel University, where he also received his PhD in Information Systems and a M.Sc. in Data Communication Systems. He has published in the areas of business process and computer network simulation. His research focuses on simulation and the evaluation of information infrastructure changes on business performance. He has a wealth of expertise gained from his work experiences in Mexico, ranging from distributed systems to computer network design. You can reach him by e-mail at <alan.edwin.serranorico@brunel.ac.uk> and his web address is <www.brunel.ac.uk/~csstaes>