

*Technical Report N° 2005/01*

*Capturing Change in Socio-technical System with  
Problem Frames*

***John Brier  
Lucia Rapanotti  
Jon G. Hall***

*4<sup>th</sup> January 2005*

---

***Department of Computing  
Faculty of Mathematics and Computing  
The Open University  
Walton Hall,  
Milton Keynes  
MK7 6AA  
United Kingdom***

***<http://computing.open.ac.uk>***



# Capturing Change in Socio-technical System with Problem Frames

John Brier      Lucia Rapanotti      Jon G. Hall  
Department of Computing, The Open University  
Walton Hall, Milton Keynes, MK7 6AA, UK  
{J.Brier, L.Rapanotti, J.G.Hall}@open.ac.uk

## Abstract

*Within organisations, business processes are increasingly captured and supported through socio-technical systems, which incorporate both people and technologies. In today's rapidly changing marketplace, such business processes are more and more complex and volatile, and so are the requirements for their supporting systems.*

*In this paper we consider change in organisations, and proposed novel techniques for the analysis of changing requirements in socio-technical systems.*

*Our approach is based on Jackson's Problem Frames, a well-known framework for the representation and analysis of software problems. We extend the framework with a notation for the representation and analysis of change for socio-technical problems.*

*We illustrate the approach on a case study from a real world situation.*

**Key Words:** *changing requirements, business process, problem domain, socio- technical system, Problem Frames.*

## 1 Introduction.

The knowledge economy [10,26] and its identification of knowledge management as a means to competitive advantage has accelerated the development of socio-technical systems - that is, systems which include both social and technological components - to support business processes. As a consequence, requirements analysis of such systems must address their real world context, and the activities and business processes through which organisations generate value [17,23]. Approaches that integrate requirements analysis with the needs for organisations have started to appear in the requirements engineering literature [3,31], although their focus has been primarily on software.

As today's businesses operate in a highly volatile often global, marketplace, dealing with change has also become an imperative, and so is the evolution of business information system [1]. Dealing with change in business processes and its impact on socio-technical requirements is the focus of this paper.

The main contribution of this paper is an approach to the representation and analysis of change in socio-technical systems in the context of organisations. The approach is based on Jackson's Problem Frames [16], and their extension to socio-technical systems in [13]. Features of the approach include a notation for the representation of change and the transition between the current situation and the changed situation within an organisation. The approach is exemplified on a real-world case study.

The paper is structured as follows. Section 2 reviews the motivation for the work and some related contributions in the literature. Section 3 gives an outline of the approach and its background. Section 4 applies the approach to the case study. Finally, Section 5 includes some reflection on the approach and concludes the paper.

## 2 Motivation and related work

Pre-1990, organisations achieved their competitive advantage by marketplace differentiation [25] (e.g., by offering competitive prices or quality for their products or services). The onset of the knowledge economy [10,26] in the early 90s, made marketplace differentiation more difficult to achieve, as information about competitors become more easily and quickly accessible through technology. Prusak [26] noted that "the only thing that gives an organisation competitive advantage - the only thing that is sustainable - is what it knows, how it uses what it knows, and how fast it can know something new". However, following the re-engineering of organisations in the 1980's and 1990's [14] (e.g., through downsizing, de-layering or outsourcing), organisations have lost some of their 'corporate memory' (e.g., what employees who had left knew but was not codified), making the

development of technology to support their knowledge management strategies more complex [10]. Inevitably, technology has also increased the pace of change in organisations.

In response, software systems have been designed for organisational management, that manipulate data, information and knowledge, with a shift from what was traditionally human activity to computer activity. This created for the software community a focus on both the development of knowledge representation [9,18,21,30] and the need to accommodate 'evolution over time' of requirements [17,23] reflecting the changing real world context.

Within the software engineering literature, the issue of change has been considered mainly in relation to software, rather than its wider socio-technical context. Among the notable contributions from the literature on software maintenance and evolution [19] are, for instance, classifications of software change [7] or management techniques for increasing software flexibility in the face of possible future software changes [2,4]. The requirements engineering literature has also addressed primarily software requirements [17]. Contributions include, for instance, classification of requirements change [22] or the relation between changing business goals and the evolution of software architecture [12].

Approaches that look at change in organisations and the impact on wider socio-technical requirements are lacking from the literature. It is in this area of research that we position our work.

### 3 Representing change for socio-technical systems

Our approach is based on Jackson's Problem Frames [16] and the extension of [13]. In this section, we briefly recall the basics of Problem Frames, and summarise the elements of our approach.

#### 3.1 Background

Problem Frames [16] are a framework for the representation, classification and analysis of software problems. (A thorough introduction to Problem Frames is beyond the scope of this paper, and can be found in [16].)

Problem Frames are a concretisation of the ideas of Michael Jackson and others in the separation of the machine and its software (the solution) and its environment's descriptions and requirements (the problem) [16]. This separation is generally accepted as being a useful principle for requirements analysis. The usual representation of the separation of machine and environment descriptions is as the 'two ellipse' model, illustrated in Figure 1.

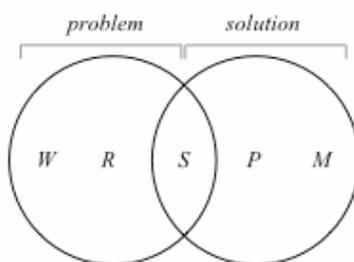


Figure 1. The requirements analysis model

In Figure 1 world knowledge *W* is a description of the relevant environment; *R* is the statement of requirements; *S* is the specification that mediates between problem and solution; *M* is the description of the machine; and *P* is the program that, on machine *M*, implements the specification *S*.

This separation is made concrete in Problem Frames through the notation of problem diagram, illustrated in Figure 2. Here a software problem is seen as that of specifying a machine (the machine domain, a rectangle marked by a double bar on the left-hand side), in a given world context (the real-world domain, a rectangle), which brings about some required behaviour (the requirement, a dotted oval). The links between domains represent their interfaces in terms of shared phenomena (e.g., states, events or commands). The requirement is expressed in terms of some real-world phenomena (*b* in the figure), while the machine specification in terms of phenomena that the machine shares with its environment (*a* in the figure).

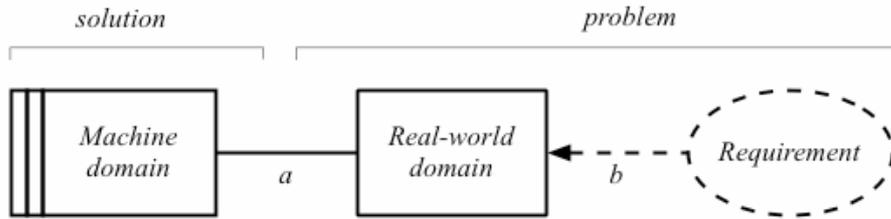


Figure 2. Basic Problem Frames notation for software problems

Note that the development process assumed by the model is one from the world ellipse (where the problem is) to the machine ellipse (where the solution is), as illustrated by the arrow in Figure 3.

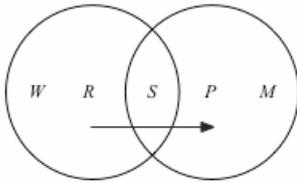


Figure 3. The development process

In [13], Hall and Rapanotti propose an extension to Problem Frames for the specification of socio-technical systems. The extension allows the design of human instruction to be the subject of the requirements engineering process addressed through Problem Frames alongside that of the machine domain. Foundationally, this means the separation of the description of the world from that of the human that is the subject of design. This leads to the reification of the original ellipse model shown in Figure 4. In it we have three ellipses – the original two, plus that for the human  $H$  with knowledge  $K$ .

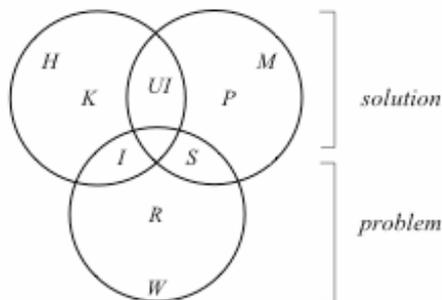


Figure 4. The socio-technical requirements analysis model

With the introduction of the human  $H$ , two new areas are identified, which now form explicit foci for design:

- the specification  $UI$ , anonymous in the  $S$  region in the original model, which determines the Human-Machine interface; and
- the specification  $I$ , missing from the original model, which determines the interaction that is expected of the human as a component of the socio-technical system.

The concretisation of this extended model in Problem Frames is the introduction of a new domain type (see Figure 5), the knowledge domain (a rectangle with a double bar on the right-hand side), to represent the human for which instruction  $K$  has to be designed.

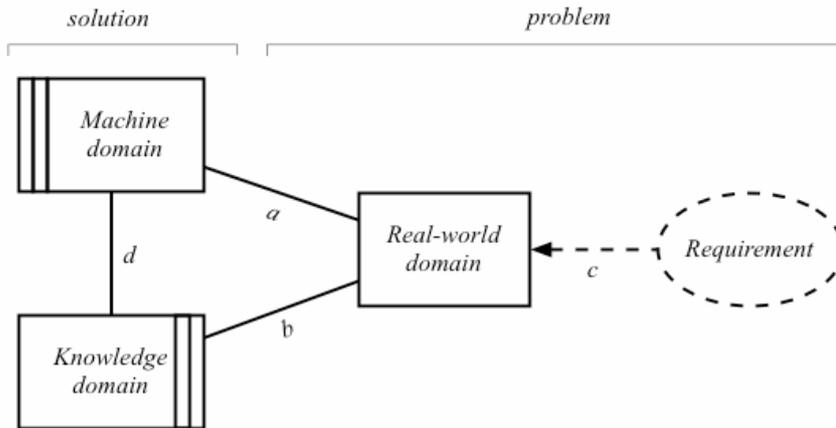


Figure 5. Extended Problem Frames notation for socio-technical problems

A solution to a socio-technical problem is then specified through the specification of both the machine and knowledge domains. The resulting development process, indicated through the arrows in Figure 6, proceeds from the world ellipse (where the problem is) to both the social and technical ellipses (where the two parts of the solution are); the social part of the solution is considered first, as this may impose further constraints on the design of technology.

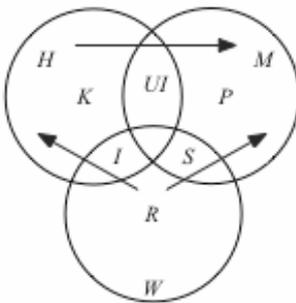


Figure 6. The development process

### 3.2 Outline of the approach

In this paper, we give an interpretation of the socio-technical model in the context of organisations with the aim of capturing changing requirements. Figure 7 illustrates this interpretation, where the human and the machine are identified with the employees and IT system of an organisation, respectively, while the world is the organisation's context of operation. Within this interpretation, we will locate change (the  $\Delta$  in the figure) and explore the effect of change in terms of the development process (the arrows in the figure).



Figure 7. An interpretation of the 3-ellipse model for organisations

In Problem Frames notation, this interpretation gives rise to the generic diagram of Figure 8.

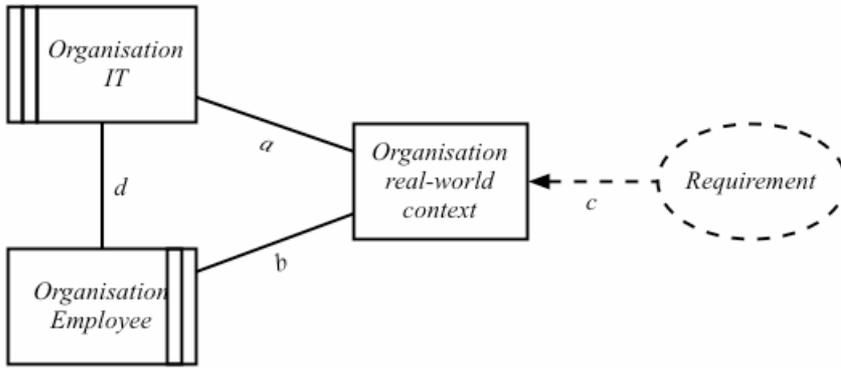


Figure 8. Extended Problem Frames notation for organisational problems

Given this interpretation, we follow the process outlined in Figure 9 for the analysis of change.

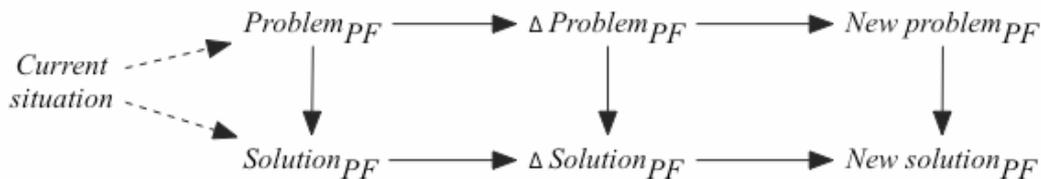


Figure 9. The change analysis process

We first capture in Problem Frames notation the current situation within the organisation (the dashed arrows in the figure). This includes the identification of the requirement which is satisfied by the current socio-technical system, together with a representation of the real-world context and the current solution. We represent this through the Problem Frames notation of problem diagrams, for which we also provide a correctness argument.

We then identify how the problem and its solution may change. To this end, we propose the notation of *change problem diagram*, as a new notation within Problem Frames, for the representation and analysis of change in problem and solution.

Finally, we represent the effect of change as a new problem and solution pair, again using the Problem Frames notation of problem diagrams, and reapply the correctness argument [16].

## 4 The Case Study

The case study is taken from [32]. Here, a major global investment bank makes use of an intermediary to handle the settlements of its customers' transactions. In turn, the intermediary, a third-party organisation, makes use of CREST, a multi-currency electronic settlement system for UK securities. Communication between the bank and the intermediary is through the bank's employees, who record customers' requests and transfer the information by phone to the intermediary. Due to increased trading volume, the bank seeks cost reduction through the elimination of the intermediary, and direct access to CREST. Through this change, the bank also aims at an improved service through customers being able to trade closer to the exchange closing time, and at an increased market credibility, as a direct participant in the UK settlement process.

### 4.1 Before the Change

A representation of the current situation in Problem Frames is given in the problem diagram of Figure 10. This represents the problem of handling customers' transactions through an intermediary, which in turns makes use of CREST to settle the transactions. In the figure, the problem context is given by the following domains and their interfaces:

- the Customer domain represents a customer of the bank;
- the Intermediary domain represents the intermediary organisation; and
- the CREST domain, represent the electronic settlement system.

The solution is a socio-technical system including the bank's IT system and employees. The Bank employees deal directly with customers and the intermediary in order to settle transactions. For brevity, we have omitted all shared phenomena.

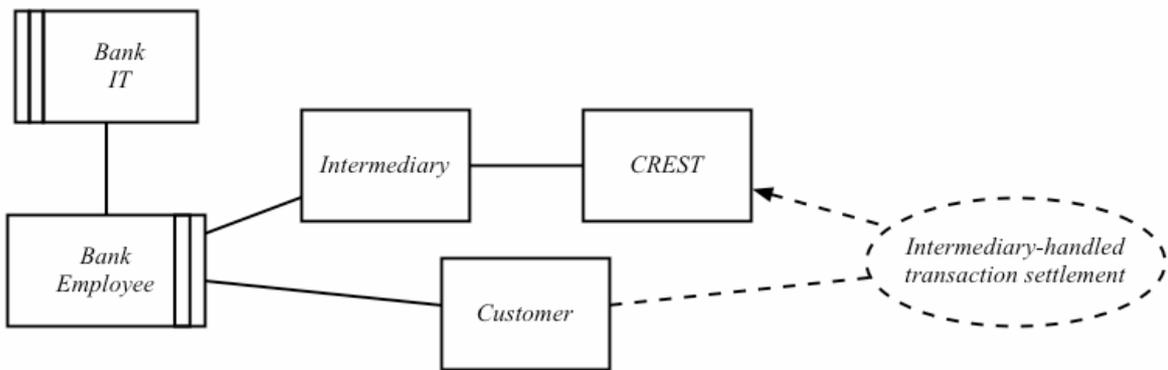


Figure 10. The problem diagram for the current situation

That the current solution satisfies the requirement is the subject of the correctness argument illustrated in Figure 11 (the argument should be followed as indicated by the numbers in the figure).

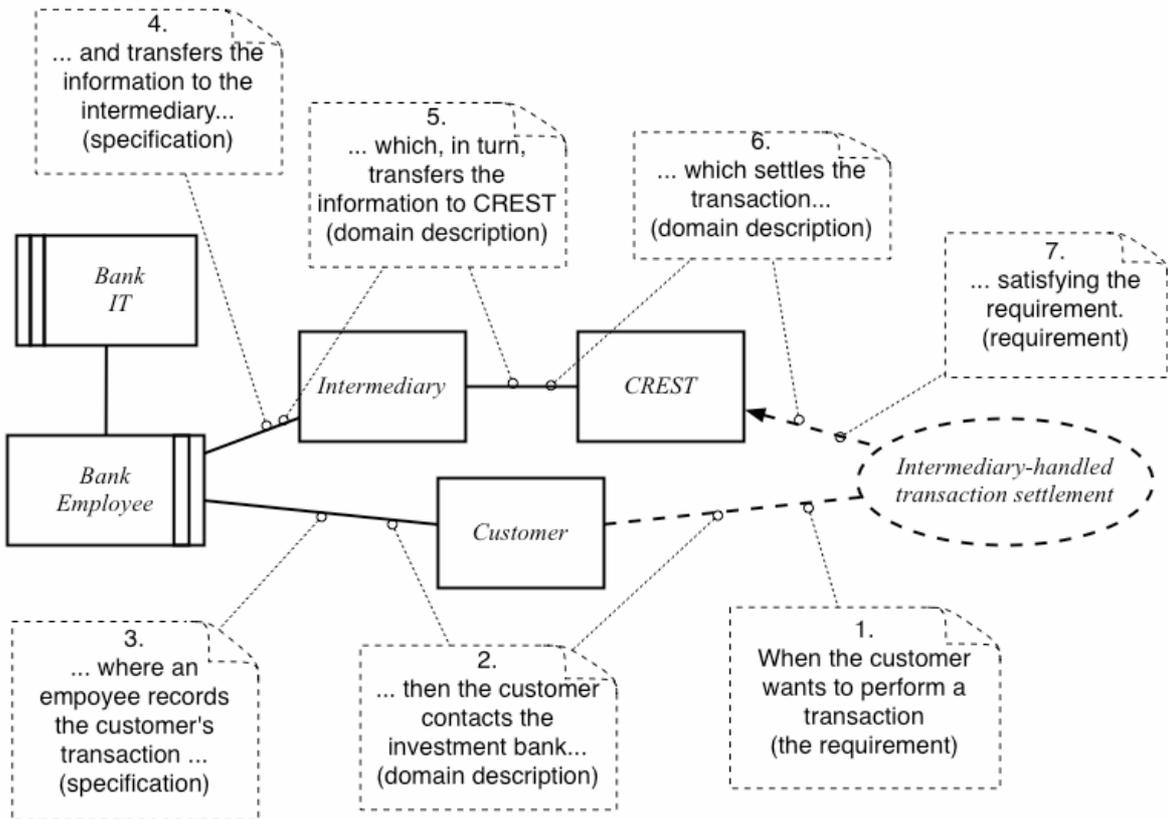


Figure 11. The correctness argument

#### 4.2 Analysing Change – the transition

As already mentioned, the bank seeks cost reduction through the elimination of the intermediary, and direct access to CREST. In order to analyse the impact of such a change on the current situation, we generate from the problem diagram in Figure 10, the change problem diagram of Figure 13, in which the impact of change is highlighted. In particular, in the figure:

- the removal of the Intermediary domain and its interfaces is represented by dotted lines;
- changes or additions in the remaining parts of the diagram are marked by a  $\Delta$ .

The changes indicate that: the Bank IT system has to be modified in order to interface CREST directly and to allow a bank employee to enter details of customers' transactions; and that the bank employee on receiving requests from a customer, must enter the information in the bank IT system, so that the transactions can be processed automatically.

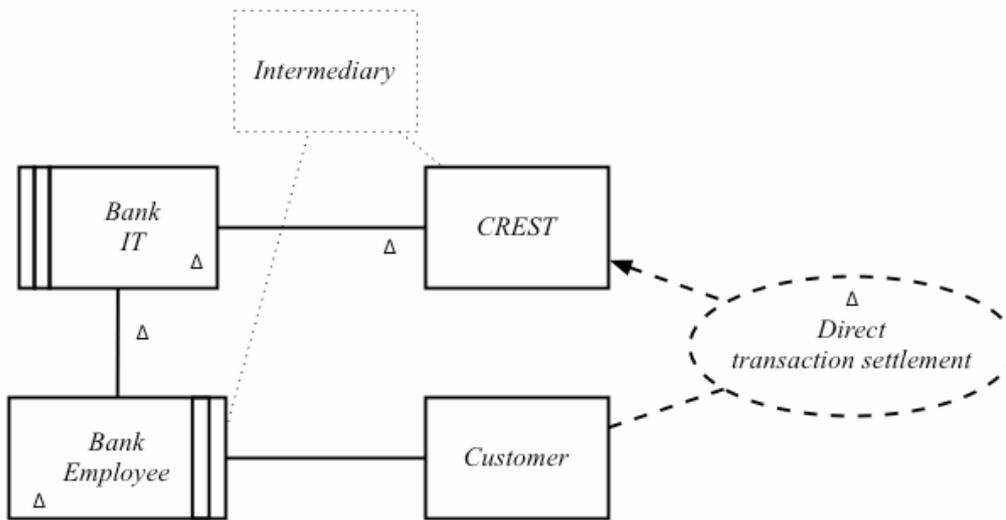


Figure 11. Change problem diagram

#### 4.3 After the change

Following the application of the changes discussed in Section 4.2, we can generate the problem diagram of Figure 12 which gives the new socio-technical solution in the face of the new requirement and context.

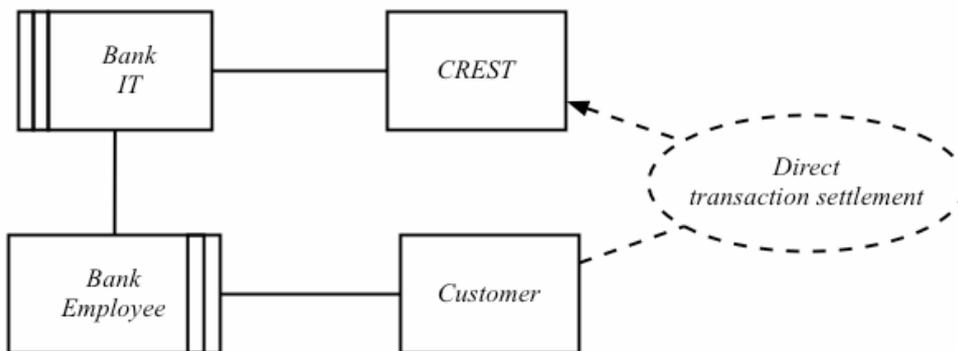


Figure 12. Problem diagram after the change

The correctness of the new socio-technical solution can then be argued through the correctness argument of Figure 13. This can be derived from that of the original problem (see Figure 11) by taking into consideration the changes to the socio-technical solution specification discussed in Section 4.2.

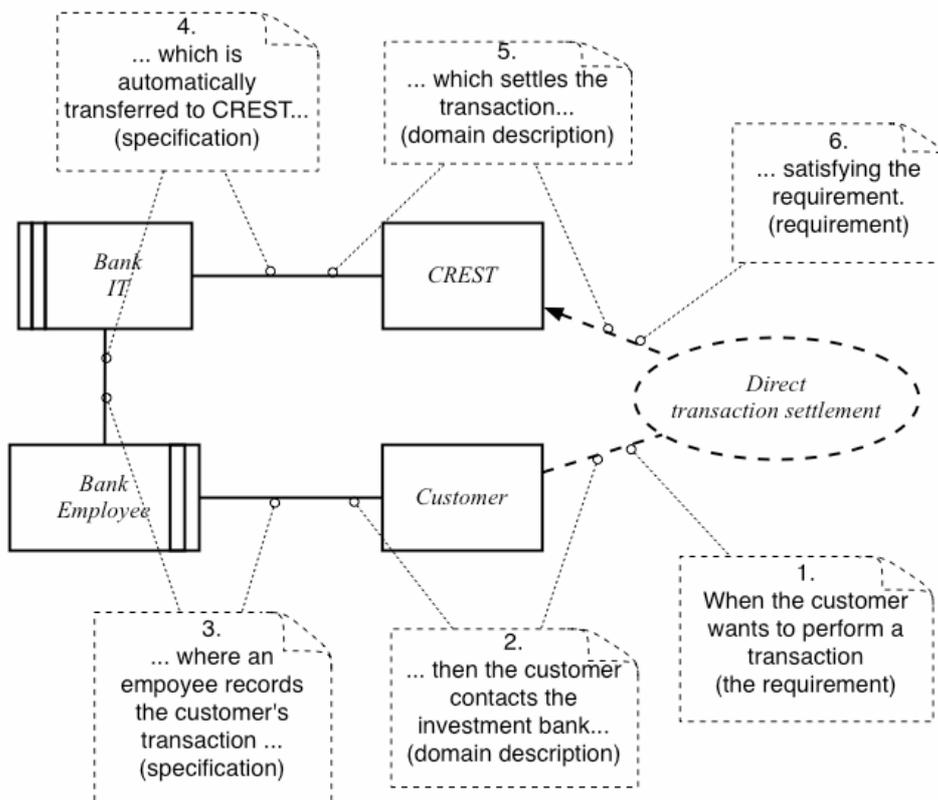


Figure 13. Correctness argument for the new situation

## 5 Discussion and conclusion

Socio-technical systems are increasingly at the heart of business organisations, supporting their main business processes. Yet, although the high pace of change in today's businesses, a comprehensive approach to dealing with change in organisations and the impact on their socio-technical systems is lacking from the literature.

In this paper we have made a first step towards addressing this issue, by proposing an approach for analysing change which is based on (extensions of) Jackson's Problem Frames. The approach has some notable characteristics. It allows for: the expression within a unified framework, of both requirements and solution specifications, and correctness statements to argue their consistency; the representation of the complex domains in which the socio-technical system operate; and the separation of social and technical parts of the solution. This is achieved through the Problem Frames notation of problem diagrams extended to socio-technical systems. It allows for the representation and analysis of change in socio-technical problems, through the new notation of change problem diagram, where change can be attributed to the separate parts of a socio-technical problem, that is its requirement, context and solution. Importantly, it includes a process for change analysis that leads from a current to a changed business situation through intermediate socio-technical problems and corresponding correctness arguments.

The approach was exemplified in this paper on a real-world case study, as a first validation of its potential in capturing change in real-world business problems.

Future work will take the approach forward in many directions. First of all, a more abstract codification of the change analysis process will be provided. Akin to the way Problem Frames classify software problems (see [16] for the codification of five such basic problems), we envisage a classification of problem changes. For instance, the cost reduction problem tackled in the case study can be seen as an instance of a more general cost reduction change where human operations within an organisation are automated - this, we know from the literature, is a very common occurrence in business. The separation of requirement, context and solution allows us to locate the different parts of the problem where change may originate. For instance, in the case study we have considered the case in which change is originated in the requirement. In general, changes which originate from the problem context or even the solution are also possible: for instance in our case study, a more competitive settlement system may become available; or the bank may need to downsize its operation and reduce personnel. Hence, we envisage a finer grain taxonomy of change, and a corresponding specialisation of the analysis process.

Finally, the example in the paper, although derived from a real-world problem, still constitutes a limited validation of the approach. We will aim at a further application of the techniques to larger real-world problems and a more extensive and systematic validation of the work.

### **Acknowledgements**

We would like to thank the following for the help advice and support we have received during the completion of this work: Michael Jackson, Bashar Nuseibeh, Zhi Li, and the Dept of Computing at the Open University.

### **References**

- [1] April A., Abran A., Dumke R.R., (2004) SM Model to Evaluate and Improve the Quality of the Software Maintenance Process.(2004) In Proceedings of the Eighth European Conference on Software Maintenance and reengineering (CSMR'04)
- [2] Bennet K et al., Panel: Software Maintenance for 2005, In Proceedings of the International Conference on Software Maintenance ICSM (1996) .
- [3] Bleistein, S., Cox, K., et al. (2004). Requirements Engineering for e-Business Advantage. Sydney, University of New South Wales.
- [4] Bohner S.A., (1996) Impact Analysis in the Software Change process: A Year 2000 Perspective: In Proceedings of the International Conference on Software Maintenance ICSM (1996)
- [5] Bohner SA., Arnold R.S., (1996) Software Change Impact Analysis, IEE Computer Society Press, Los Alamos, California.
- [6] Brier, J., Rapanotti, L., et al. (2004). Problem Frames for Socio-Technical Systems: predictability and change.In 1st International Workshop on Advances and Applications of Problem Frames, Edinburgh, IEE.
- [7] Buckley J., Mens T., Zenger M, Rashid A., Kniesel G., (2002) Towards a taxonomy of software change, Journal of Software Maintenance and Evolution: Research and Practice
- [8] Chudge,J. Fulton,D., (1996) Trust and co-operation in system development: applying responsibility modelling to the problem of changing requirements. In Software Engineering Journal, Vlume 11, Issue 3, pages 193-204.
- [9] Dervin B (1999) Chaos order and sensemaking; in Jacobson, R (ed) Information Design, Boston, Mass. MIT Press.
- [10] Gibbons M (1994) The New Production of Knowledge-Sage Publications, London
- [11] Gunter, C. A., Gunter, E. L., Jackson, M., Zave, P. (2000). "A reference Model for Requirements and Specifications." IEEE Software 17(3): 37-43.
- [12] Gross,D.Yu,E.(2001) Evolving System Architecture to Meet Changing Business Goals: an Agent and Goal-Oriented Approach, In Proceedings of the Fifth International Symposium on Requirements Engineering (RE,01)
- [13] Hall J.G and Rapanotti L. (2005). Problem Frames for Socio Technical Systems. In J. Mate and A. Silva editors, Requirements Engineering for Socio-technical Systems, Idea Group Inc. To appear.
- [14] Hammer. M (1990) Re-engineering work: don't automate, obliterate', Harvard Business Review, July-Aug.,pp.104-12.
- [15] Harker.S.D.H., Eason,K.D., Dobson, J.E. (1993)The Change and Evolution of requirements as a Challenge to the Practice of Software Engineering, In Proceedings IEEE International Symposium Requirements Engineering, San Diego, CA, USA.

- [16] Jackson. M.A. (2001) Problem Frames. Addison – Wesley,
- [17] Jarke, M. Pohl, K., (1994) Requirements Engineering in 2001 (virtually) managing a changing reality, In Software Engineering Journal, pages 257-266, vol.9 issue 6.
- [18] Jones A., (1991) Logic and Knowledge Representation Pitman Publishing.
- [19] Katayama T., Tamai Tetsuo., Yonezaki N., (2000) Principles of Software Evolution Proceedings of the International ISPSE 2000 Japan. IEEE inc.
- [20] Lavazza, L. and Del Bianco, V. (2004). A UML-based Approach for Representing Problem Frames. In 1st International Workshop on Advances and Applications of Problem Frames, Edinburgh, IEE.
- [21] Nonaka, I. and Takeuchi, H. (1995) The Knowledge-Creating Company, Oxford University Press.
- [22] Nurmaliani N., Zowghi D., and Williams S., "Using Card Sorting Technique to Classify Requirements Change", Proceedings of the 12th IEEE International Requirements Engineering Conference (RE'04), Kyoto, Japan, September 2004
- [23] Nuseibeh B. Easterbrook S. (June 04) Requirements Engineering: A Roadmap. In The Proceedings of the conference on the Future of Software Engineering p.35-46.
- [24] Patel, N.V. (1999) The Spiral of Change Model for Coping with Changing and Ongoing Requirements, In Requirements Engineering, Volume 4, No2, pages 77-84.
- [25] Porter M. E., (1985) Competitive Advantage: Creating and sustaining superior performance, Free Press.
- [26] Prusak, L., (1996) 'The Knowledge Advantage, Strategy and Leadership, Vol.24, No.2, pp.6-8.
- [27] Rapanotti, L., Hall, J.G., Jackson, M.A., and Nuseibeh, B. (2004) Architecture Driven Problem Decomposition. In Proceedings of the 12<sup>th</sup> IEEE International requirements Engineering Conference (RE'04) Kyoto, Japan. IEEE.
- [28] Stacey R.D., (2003) Strategic Management and Organisational Dynamics-Prentice Hall.
- [29] Strens, M.R., Sugden, R.C., (1996) Change Analysis: A Step towards Meeting the Challenge of Changing Requirements, In Proceedings of the IEEE Symposium and Workshop on Engineering of Computer Based Systems (ECBS'96) Germany.
- [30] Weick, K. E. (1995) Sensemaking in Organisations, Thousand Oaks, Calif. Sage Publications.
- [31] Weiringa, R., Gordijn, J., et al. (2004). Value Framing: A Prelude to Software Problem Framing. 1st International Workshop on Advances and Applications of Problem Frames, Edinburgh, IEE.
- [32] [www.katotech.co.uk](http://www.katotech.co.uk). Last visited: 29<sup>th</sup> November 2004.