Design Rationale Capture with Problem Oriented Engineering: an Investigation into the Use of the POE Framework for the Capture of Design and Architectural Knowledge for Reuse within an Organisation

A. Nkwocha

28 June, 2008

Department of Computing
Faculty of Mathematics, Computing and Technology
The Open University

Walton Hall, Milton Keynes, MK7 6AA
United Kingdom

http://computing.open.ac.uk
Design Rationale Capture with Problem Oriented Engineering: An investigation into the use of the POE framework for the capture of design and architectural knowledge for reuse within an organisation.

A dissertation submitted in partial fulfilment of the requirements for the Open University’s Master of Science Degree in Software Development

Ann Nkwocha
(X5241342)

10 March 2009

Word Count: 15,507
Preface

Furthering my academic career has long been a desire of mine but a busy job and large family have made it difficult to achieve in the past. Being able to complete this research is a realisation of a long held ambition. It is my hope that my achievements also serve as an encouragement to others to strive hard to reach their goals.

I would like to take this opportunity to acknowledge all those whose assistance and encouragement helped me through this research. In particular I would like to thank:

My employers who contributed towards my studies and especially my boss Mr Bob Crafer who gave me lots of encouragement and the time I needed to accomplish tasks.

Mr Specialist advisor Dr Jon Hall whose wealth of knowledge and well grounded advice kept me on track to complete this research.

My husband Gus for his attention to detail in proof reading my work who along with my children, Annette, Daniel, Jonathan and Ian had to share me with my academic venture.

My mum Olivia Ogo who’s cooking very often kept my family from going hungry while I worked away at assignments.

Ted Fisher, my tutor, for his constructive comments on my assignments.

And all my friends and family whose kind words and support have helped me see this research through to the end.

Thanks to you all.
Table of Contents

Preface ........................................................................................................................... i
List of Figures ................................................................................................................. vi
List of Tables .................................................................................................................... viii
Chapter 1  Introduction .................................................................................................. 10
  1.1  Background to the research .................................................................................. 10
    1.1.1  Design Rationale and the Importance of Rationale Management .................. 11
    1.1.2  Capturing Design Rationale – Methods, Types and Guidelines .................. 13
    1.1.3  Problem Oriented Engineering ....................................................................... 15
  1.2  Research Aim and Question .................................................................................. 17
  1.3  Objectives of the Research ................................................................................... 18
  1.4  Dissertation Overview ........................................................................................ 20
Chapter 2  Literature Review ......................................................................................... 21
  2.1  Research Context .................................................................................................. 21
    2.1.1  Design rationale - Background and Current Research ................................ 21
    2.1.2  Problem Oriented Engineering (POE) .......................................................... 27
    2.1.3  Summary ......................................................................................................... 32
  2.2  Intended Contribution to Research ....................................................................... 35
Chapter 3  Research Methods ......................................................................................... 37
  3.1  Research methods considered .............................................................................. 37
    3.1.1  Prototype and Test .......................................................................................... 37
3.1.2 Document Studies

3.1.3 Case Studies

3.2 The Selected Approach – Single Case Study

3.2.1 Case Study Design

3.3 Threats to validity and ethical considerations

Chapter 4 Data Collection

4.1 Introduction

4.1.1 Scope of the Study

4.2 The Questionnaire

4.3 Document Sources

4.4 Defining the Valid Amounts Case Study

4.4.1 Overview

4.4.2 Handling Problems in the organisation

4.5 Applying POE to the Valid Amounts Problem

4.5.1 POE Notation and Diagrams

4.5.2 Abbreviations

4.5.3 The POE Role Assignments

4.5.4 Problem Exploration

4.5.5 Problem Validation

4.5.6 Solution Exploration

4.5.6.1 The Mortgage Processing (Calculations) Sub-Problem
4.5.6.2 The Loan Balances Sub-Problem ............................................................. 66

4.5.7 Solution Validation .................................................................................... 67

4.5.8 Progressing the Loan Balances Solution ................................................ 72

4.6 Summary ........................................................................................................ 74

Chapter 5 Results ................................................................................................. 75

5.1 Results from the Questionnaire ................................................................... 75

5.2 Case study Results - Design rationale Elements Captured using POE .......... 79

5.2.1 Decisions .................................................................................................. 80

5.2.2 Rationale for Decisions ............................................................................ 80

5.2.3 Alternatives and Trade-offs ................................................................. 80

5.2.4 Suitability ............................................................................................... 81

5.2.5 Constraints .............................................................................................. 81

5.2.6 Assumptions ............................................................................................ 81

5.2.7 Status ....................................................................................................... 82

5.2.8 Complexity of Design ............................................................................ 82

5.2.9 Issue Descriptions .................................................................................. 82

5.2.10 Contacts ................................................................................................ 82

5.3 Reusing Captured Design Rationale ........................................................... 83

5.4 Summary ........................................................................................................ 86

Chapter 6 Conclusions ......................................................................................... 89

6.1 Project review .............................................................................................. 89
Design Rationale Capture with Problem Oriented Engineering: An investigation into the use of the POE framework for the capture of design and architectural knowledge for reuse within an organisation.

6.1.1 Methodology .................................................................90

6.1.2 Achievement of Research Aim and objectives..........................91

6.2 Discussion ........................................................................93

6.3 Recommendations for Future Work .........................................95

References ...........................................................................97

Bibliography ........................................................................101

Appendix A - Questionnaire ......................................................104

Appendix B – POE Notation and Templates .................................113

Appendix C – Case Study Details ...............................................125
List of Figures

Figure 1 POE Process Pattern .................................................................31
Figure 2 Flow of issue information ..........................................................47
Figure 3 Design tree for the Valid Amounts Problem ..................................50
Figure 4 - The Valid Amounts Problem .......................................................54
Figure 5 Partial Candidate Problem Exploration Step as a Problem ...............57
Figure 6 Problem Validation Triage Document ..............................................58
Figure 7 - Problem P1 Expanded Valid Amounts Domain ................................60
Figure 8 Problem transformation – P1 to P2 with Validation ............................63
Figure 9 Mortgage Processing Architecture ..................................................66
Figure 10 Loan Balances Sub-Problem ........................................................67
Figure 11 Valid Amounts Solution Validation .................................................71
Figure 12 Expected Values Architecture .......................................................73
Figure 13 Adjustments .................................................................................73
Figure 14 Expanded Valid Amounts Solution ..................................................74
Figure 15 - Information Helpful for Software Evolution ..................................76
Figure 16 Sources of Information about Software ...........................................77
Figure 17 Factors that can Affect Design Rationale Reuse ...............................78
Figure 18 Factors that affect Design Rationale Capture ....................................79
Design Rationale Capture with Problem Oriented Engineering: An investigation into the use of the POE framework for the capture of design and architectural knowledge for reuse within an organisation.

Figure 19 Flow of Information Revised with the POE process ...........................................85
Design Rationale Capture with Problem Oriented Engineering: An investigation into the use of the POE framework for the capture of design and architectural knowledge for reuse within an organisation.

List of Tables

Table 1 Elements of Design Rationale identified from Literature Survey......................34
Table 2 Description of Terms.........................................................................................43
Table 3 POE Diagram Symbols ....................................................................................51
Table 5 Abbreviations ..................................................................................................52
Table 6 POE Role Assignments....................................................................................53
Table 7 Elements of Design Rationale Captured in POE Process Execution.................87
Abstract

Design rationale in software engineering fills in the gaps between the original requirements of a system and the finished product encompassing decisions, constraints and other information that influenced the outcome. Existing research in this field corroborates the importance of design rationale for the evolution of existing systems and creation of new systems. Despite this, the practice of design rationale capture and reuse is not as extensive as could be expected due to reasons which include time and budget constraints and lack of standards and tools.

This capture of Design Rationale during software design activities carried out using Problem Oriented Engineering (POE) was demonstrated with the use of a case study. POE is a formal system for engineering design that provides a framework for the resolution of software problems in a stepwise manner.

A review of literature on Design Rationale, its capture and management yielded a list of elements used as the criteria for identifying design rationale in the information gathered during the case study. Examination of that information revealed that all the identified elements were recorded and led to the conclusion that Design Rationale is captured when solving a software problem using POE.

Examination of the flow of information that occurred during the execution of the case study led to the conjecture that Design Rationale recorded during the case study could be reused. Successful reuse would, however depend on the effectiveness of the categorisation, storage and organisation of the information gathered.
Chapter 1  Introduction

This chapter contains background information to the research undertaken. It contains an overview of Design rationale and the challenges associated with its capture. Problem Oriented Engineering (POE) framework is outlined and the chapter concludes with the aims and structure of the project.

1.1 Background to the research

Computers and software are ubiquitous in present day society of the western world. They underpin business, the economy and economic trends, careers, health and welfare to name a few. Computers and software are constantly changing in the attempt to keep pace with demand, meet new needs and exploit new technologies.

With so much invested in software, there is a necessity to guard the knowledge and experience necessary to allow for its evolution. While some organisational knowledge is tangible and embodied within its software systems and related artefacts, a large part of it remains tacit (i.e. within the memory of individuals) if steps are not taken to externalise the knowledge. Subject matter experts and original development team members for example may have information on the reasoning behind decisions taken and/or discarded designs that are not recorded anywhere else. This knowledge and experience can then be lost through staff movement and attrition. (M882 2006a)

In the fast paced global economy within which most companies must operate, an organisation's software must evolve if it is to keep up with changing business needs. These needs are influenced by the human and social domains in which the software is used.
Expertise is achieved as a result of dedicated application to a chosen field, having been exposed to many examples of the problems and solutions that occur in their domain (Cross, 2004). In a corporate setting, knowledge has been seen as a strategic factor in gaining competitive advantage in today’s knowledge based society (Xin et al., 2001). The ability to harness and use expert knowledge can therefore contribute to the continued success of an organisation.

Traceability from requirements for the software to final software product can provide beneficial input to the activities involved in evolving systems (M882, 2006b). This is however, difficult to achieve due to the differences in levels of abstractions and notations used for requirement and software component artefacts. In addition to this, in many cases the documentation is not kept up to date with changes made after the initial requirements are finalised.

1.1.1 Design Rationale and the Importance of Rationale Management

Design rationale bridges the information gap between the requirements of a system and the final design described as a snapshot of final design decisions (Burge et al., 2000). It captures the reasoning and knowledge that justify the resulting design (Tang et al., 2006). This knowledge includes arguments behind choices taken that resulted in these final decisions.

There are many reasons that make the availability of design rationale of benefit to organisations. One that has become of more relevance has been precipitated by the emergence of the global software development team which has presented several challenges to the software development process. The need to support outsourced or
dispersed development teams necessitates the development of new and extension of existing techniques for requirements engineering (Cheng et al., 2007). Further challenges posed by the globalisation of software engineering projects include communication concerns within teams which may consist of geographically dispersed members and the likelihood of further evolutions of the software managed globally.

A Gartner report on IT services (Beck, 2002) projected that nearly half of Fortune 1000 global enterprises will choose to draw business benefits from shared IT utility infrastructures owned and operated by service provider hybrids rather than own their IT assets.

The transfer of tacit knowledge held by software developers in conditions of time, space and cultural specifications is not without its problems and the effect of staff attrition on the offshore software development process is a very significant parameter for success (Mathrani et al., 2005)

These factors re-emphasise the need for effective management knowledge especially where expertise may be lost due to staff attrition which is a common consequence of off-shoring and/or outsourcing programmes.

Despite the highlighted importance of design rationale, it is still not always captured for reasons which include the fact that it can be time consuming and expensive (Burge et al., 2000) and often places extra burdens on resources facing tight deadlines without presenting any immediate benefits to them (Krutchen et al., 2005). In a survey of design practitioners, lack of standards, time and budget constraints and lack of tools to support the capture are among reasons cited for not documenting design rationale (Tang et al., 2006)
1.1.2 Capturing Design Rationale – Methods, Types and Guidelines

In an article surveying design rationale systems five methods of capturing design rationale are described (Lee, 1997). These are (listed in order of complexity)

- Reconstruction - design rationale is captured in a raw form using techniques such as video and interviews and then translated into a more formal structure
- Record and replay - here rationale is captured as it occurs during design sessions such as video conferencing and e-mail bulletins
- Methodological by-product - Uses a schema based approach to capture design rationale during the process of design
- Apprentice - uses a rich knowledge based approach to question actions taken by the designer
- Automatic generation - design rationales are generated automatically from a Artificial learning based compilation of execution history

Crucial to the effective capture of rationale is the choice of elements to be recorded. An empirical survey of design rationale carried out was based on 9 generic areas of design rationale typically captured during the development process (Tang et al., 2006). These are:

- Design constraints
- Design assumptions
- Weakness of a design
- Benefit of a design
- Cost of a design
• Complexity of a design

• Certainty of design

• Certainty of implementation

• Tradeoffs between design alternatives.

The representation of information gathered as design rationale also features in research as this will determine how accessible the information will be. Research in this area centres around models of argumentation which provide a framework for the representation of design ideas and their relations (Jarczyk et al., 1992).

The recognition of the importance of documenting and managing rationale has led to the emergence of industry guidelines aimed at standardising elements and practices to provide a basis to improve cost efficiencies and quality (Tang et al., 2006). These include IEEE Recommended Practice for Architectural Description of Software-Intensive Systems (IEEE 1471/2000 standard) and Views and Beyond approach to documenting Software Architecture (V&B) guidelines.

The IEEE Recommended Practice for Architectural Description of Software-Intensive Systems (IEEE 1471/2000 standard) was formulated by the IEEE Architecture Planning Group (APG) whose goal is to set a direction for including architectural thinking into IEEE standards. The standard provides guidelines for recommended practice in creating, analysing and maintaining architectures of software-intensive systems and the documenting such architectures in terms of architectural descriptions. The standard sets out a conceptual framework for architectural description of software intensive systems and characterises the content of an architectural description.
The V&B approach - Documenting Software Architectures: Views and Beyond was created by Carnegie Mellon Software Engineering Institute (SEI) offering both a documentation philosophy and a detailed approach for documenting system architecture. It characterises the documentation software architecture as a matter of choosing a set of relevant views of the architecture, first recording each of the chosen views then the information that applies to the whole set of views or individual ones. Finally the views are bound together in a way that presents an interdependent and integrated representation of the architecture. The approach includes a method to choose the most relevant views, standard templates for documenting a view and documenting the information beyond views, and definitions of the templates’ content (Clements, 2005)

1.1.3 Problem Oriented Engineering

Problem Oriented Engineering POE is a formal system for engineering design that provides a framework for the resolution of software problems in a stepwise manner (Hall et al., 2006). Problems are transformed through a series of steps to reach a solution to a stated requirement. Originally presented as Problem Oriented Software engineering - POSE, the framework is the result of research extending Problem Frames, aimed at analysing and structuring problems, to include activities addressing the solution domain.

The framework has been enhanced by further research resulting in the inclusion of architectural patterns, adequacy arguments for justifying design decisions taken and provided a process map for the application of the framework to the development of safety critical systems (Hall et al 2007 a).
While early research focused on the software engineering arena, the framework was subsequently abstracted for application to engineering in general (Hall et al 2007 b). This research focuses on the application of POE in a software engineering context using the generic process pattern of the framework. This will encompass references to research done specifically for software engineering.

The POE approach seeks to bring formal and non-formal aspects of the development process together (Hall et al., 2006).

The application of POE in a case study to solve a package router problem concluded that the concept of problem solving in POE was not biased to a software solution inferring a potential extension to other areas of problem solving (Hall et al 2007 a). This premise was explored and led to the POE process pattern being further enhanced and generalised. Extensions have been added to facilitate documentation and analysis of rationale for decisions; the ability to explicitly consider design risks; rich traceability between requirements and design elements of the framework and; the ability to parameterise problems for diverse use (Hall et al., 2007 b).

The outlined characteristics of POE suggest that it would be effective for the use of design rationale capture. However, the benefits of capturing design and architectural knowledge are only realised when the information can be accessed and used to assist in further developments, decisions and learning within an organisation that has invested in the practice. Value is achieved in the use of the captured rationale (Burge et al., 2000).
More recent research has sought to apply Problem Oriented Engineering principles outside of software engineering (Hall and Rapanotti, 2008a). A demonstration of the use of POE in the publishing domain was carried out with the intent to show that the framework could be applied in a socio-technical setting to capture knowledge. Hall’s paper puts forward the argument that the application of POE to transform problems to solutions is in fact a process of capturing knowledge.

POE’s ability to classify problems within domain contexts and the traceability that is inherent in the framework are features which could make it possible to access information to support further design and decision making activities.

1.2 Research Aim and Question

This research aims to show how design rationale capture can be achieved during the design process. The POE process will be used to design a solution for a software problem to demonstrate this premise.

To accomplish this aim, the following question will be addressed:

*What information gathered during the application the POE framework to a software design problem makes it suitable to represent the design rationale of the problem in a form that facilitates future re-use?*

The investigation will consider the following points.

- Whether the information captured is sufficient to represent design rationale
- Whether the information that is captured has a structure that will allow it to be accessed for future reuse
1.3 Objectives of the Research

In order to answer the main research question, the following research objectives are proposed:

1) Investigate through a survey of available literature, current research into design rationale and its capture.

2) Identify from current research and industry guidelines, key data elements that should be captured as design rationale and that facilitate its management and reuse.

3) Investigate the views of typical IT project and support staff on the availability of design rationale in the candidate organisation, its capture and usefulness.

4) Examine, through the use of a case study, which elements of design rationale identified in the literature review are captured by the application of POE to solve a software engineering problem.

5) Explore the structure of the data captured by POE to assess how it may be accessible for reuse.

6) Summarise the results of the research and propose recommendations for future developments and/or further research.

Literature in the field of Design Rationale capture suggests that, while the benefits of the capture and use of design rationale is widely accepted, more research is needed on what constitutes essential elements of design rationale and support mechanisms to facilitate their capture (Tang et al., 2006).

A major obstacle to the practical capture and reuse of design rationale within organisations is the fact that recording design decisions is seen as a cumbersome and tedious overhead with no immediate benefits to the practitioner. It is commonly
considered to be just another form of documentation and is viewed as expendable if deadlines for deliveries are threatened (Burge et al., 2000). Improvements in this area may be achieved by demonstrating to design practitioners that rationale management can be achieved without the perceived tedium and with identifiable benefits in order to encourage its uptake.

Existing research on the topic of problem oriented research asserts that the capture of design decisions happens within the application of the process (Hall and Rapanotti, 2008a). This implies that design rationale capture can be achieved as a methodological by-product of using the POE process pattern for design. An anticipated outcome of this demonstration is that it will contribute to addressing concerns over cost effectiveness of design rationale capture (Lee, 1997).

A survey of problem frames research (Cox et al., 2005) highlights the need for realistic case studies to demonstrate the practical use of problem frames (in this case the derivative form - Problem Oriented Engineering) within industry settings. The case study chosen for this research is a real life issue within a company developing and maintaining software for the mortgage industry.

Globally distributed development teams and stakeholders are becoming more widespread in the computing industry (Cheng, 2007) and the transfer of IT roles to offshore resources often lead to the loss of experienced resources in the on-shore organisation. In those environments, ensuring the capture and retention of expertise for future use is crucial.

In a paper addressing knowledge within organisations, architecture and design knowledge which encompasses the knowledge and expertise its employees is described
as part of corporate memory of the organisation (Alvarado et al., 2004). Efficient classification of information and easy methods to create navigate and retrieve organisation memory are necessary for benefits to be achieved.

While the production of a system to automate these functions using the POE framework is beyond the scope of this project, it is hoped however, that this research will add value by assessing the how the information collected during the use of POE for can facilitate creation of and access to design rationale.

1.4 Dissertation Overview

The rest of this research report is ordered as follows:

Chapter 2 presents a review of literature in the areas of design rationale and Problem Oriented Engineering. From existing research, criteria for assessing the information captured by POE are outlined. Chapter 3 outlines the research strategy adopted and how it supports the aims and objectives for this research. Chapter 4 describes data collection activities and the case study while Chapter 5 examines the information gathered from case study and other sources. Finally, Chapter 6 draws conclusions from the research and outlines potential opportunities for future work.
Chapter 2  Literature Review

This chapter reviews existing literature on Design rationale, capturing design knowledge and Problem Oriented Engineering.

2.1 Research Context

This literature review was undertaken to assist in learning about my chosen topic area and to describe research that has been undertaken in the areas of design rationale, its capture and management and Problem Oriented Engineering.

For clarity and to ease understanding, the review is organized in the following themes

- Definitions of and research into Design rationale its capture and management
- Problem oriented engineering

2.1.1 Design rationale - Background and Current Research

In a survey of design rationale in the software engineering field (Jarczyk, 1992), capturing design rationale is described as particularly important for aiding the development, evolution and support of large software systems whose life cycle would usually involve a large number of programmers, system analysts, project and section leaders and other support staff. The knowledge needed to support the system remains spread over a large number of people and is lost as they move on to other projects or organisations. In order to support the communication and reuse of the design knowledge it must first be captured (Hu et al., 200).
While capturing the knowledge is a crucial step, the value of the knowledge assets of an organization are actually realised when the captured knowledge is available in a usable form when it is needed in the future (Canas et al., 1999; Burge et al., 2000).

To exploit architectural knowledge either for the evolution of a system or development of a new one, information embodied in the architecture design is insufficient (Kruchten et al., 2005). The reasoning, constraints and other factors that affected the design decisions taken are also important. This information or “rationale” of the resulting software product often remains tacit and is often lost when the individuals in possession of knowledge move on. (Hu et al., 2000).

Design Rationale in its simplest form is the “…the explicit listing of decisions made during a design process and the reasons why those decisions were made” (Jarczyk et al., 1992). Research into design knowledge and rationale indicates that the knowledge and reasoning that justify the resulting design includes how it satisfies functional and quality requirements, the reasoning that caused design choices to be made over other options and what type of system behaviour is expected under different environmental conditions (Gruber et al., 1991; Lee, 1997).

Despite the benefits afforded by the availability of design rationale, research into the area highlights several issues that hinder its capture in practice. These include:

- How to recognise what constitutes a design decision and how and when they should be captured (Lee, 2007; Tang et al., 2005; Jarczyk et al., 1992).
- How architects and designers decide what is a reason or intention with regards to the decision they have made? (Jarczyk et al., 1992)
What form the representation of the design rationale should take (Lee, 1997; Douglas, 2005)

The process of recording all decisions made and those rejected requires additional time and expense which quite often cannot be afforded on projects. Consequently practitioners become resistant to the process if they feel it is too intrusive (Burge et al., 2000).

The time and budget constraints to capturing Design Rationale was found to be the most common barrier to documenting rationale in a survey of design practitioners and their perception and use of Design Rationale DR (Tang et al., 2006).

A lack of tools to facilitate the capture of design rationale is cited as a hindrance to recording the necessary information to the practice (Tang et al., 2006).

Research into capturing design rationale (Douglas, 2005) states that rationale management should start with the examination of the problem to be solved and should include

- A description of issues addressed prior to the decision.
- A list of the alternative solutions considered.
- The criteria used in the selection.
- The argument or reasoning used to justify each alternative.
- The decision itself
Research undertaken by Lee (1997) identifies the main issues that must be addressed by a design rationale system to establish what should be captured and when. Issues to be addressed are listed as:

1. What services to provide – this is determined by what the system will be designed to support – learning, documentation, better design or better maintenance.

2. What to represent explicitly – not all rationale can be captured explicitly and what is captured must be accessible. The paper presents a generic representation in the form of layers of information.

3. How to represent the rationale – this is dependent on the system representation notation. The degree of formality of the representation guides how automated the capture and retrieval of rationale can be.

4. How to produce the rationales – 5 methods of capturing design rationale are described - listed in order of complexity as
   a. Reconstruction
   b. Record and Replay
   c. Methodological By-Product
   d. Apprentice
   e. Automatic generation.

5. How to access rationales – 2 main retrieval methods described are “user initiated” and “system initiated”.

Design Rationale Capture with Problem Oriented Engineering: An investigation into the use of the POE framework for the capture of design and architectural knowledge for reuse within an organisation.
6. How to manage rationales cost effectively – the systems will not be used if the cost (the cost of producing the rationale in reusable form) outweighs the perceived benefits (the services the system will provide).

7. How to integrate the system – making the design rationale system part of design activities

Lee concludes that more research is required in the areas of designing cost effective systems and providing methods to produce formal design rationales. The capture of design rationale by Methodological By-Product may address the cost issue as additional time and costs are minimal. POE claims to capture design rationale as part of the process of its execution (Hall et al., 2007b) a claim that will be examined in this research project.

The knowledge assets of an organisation include the tacit knowledge of experienced task experts (Canas et al., 1999). In order to be reused, it must be extracted from these individuals and then recorded in a way that can be accessed and used by others.

Researchers (Jarczyk, et al 1992; Lee, 1997; Hu et al., 2000) have observed that the representation of design rationale is a significant factor in its reusability.

Hu et al (2000) identify two main approaches as having emerged for the development of systems to aid the capture of design rationale. Process-oriented approaches are more often used where there may not be established design principles and captures the rationale as a history of the system being examined. Feature-oriented approaches focus on the representation of artefacts in the system and the rules constraining the design process more suited to environments with a high degree of standardisation.
The choice of representation schema (how the rationale will be stored and presented for retrieval) is identified as being of critical importance of effective design and reuse of the captured information. Supporting communication and reuse of the design knowledge is named as the primary requirement for its capture (Hu et al., 2000) and capture can be implemented through user intervention (where practitioners capture the data), automatically or a combination of both.

Burge and Brown (2002) discuss how the use of a process model can assist in guiding decisions that need to be made during the design activity and later provide knowledge that can be used to make decisions. They suggest that the explicit capture of alternatives considered and the rationale behind the choices made can be integrated into a design process model. The design process could possibly be repeated to create a new but similar design or to assess the impact of changing decisions taken during the original design.

The benefits of having recorded knowledge are realised if it can be leveraged subsequently for decision making. Farenhorst et al., (2007), propose a feedback structure for sharing architectural knowledge using incentives to promote its practice. They list the incentives as

- Establishment of social ties
- More efficient decision making
- Knowledge internalisation.
2.1.2 Problem Oriented Engineering (POE)

Burge and Brown (2002) define a design process as: "... the set of steps, or activities, that take place in achieving the design goals, or objectives. Models of the design process are used in order to either describe the activities of the design process or prescribe how the designing should be done. Many decisions need to be made while designing. A process model can assist in guiding what decisions should be made when, and if, the model describes the design of a specific artefact, can even provide the knowledge to be used to make the decisions."

Hall et al (2006) define Problem Oriented Software Engineering (POSE) as a process for software development that seeks to bring formal and non-formal aspects of the development activities together. The process extends problem frames from the problem to the solution space with the intent of providing a conceptual framework for software development.

Software problems in POSE are defined as having a Real World Context, a Requirement and a Solution. The real world context consists is a collection named descriptions, called domains; and associated phenomena (events, commands, states etc.). Requirements are also domains with a name descriptions and phenomena expressing a “wish” to be satisfied and communicate how a proposed solution description can be validated as suitable for solving the problem. In other words, the solution will solve the problem if it satisfies the problem requirement. Solutions are designed using stepwise transformations of software problems.

The framework allows for:
Design Rationale Capture with Problem Oriented Engineering: An investigation into the use of the POE framework for the capture of design and architectural knowledge for reuse within an organisation.

- Identification and clarification of system requirements
- Understanding and structuring of the problem world,
- The structuring and specification of a hardware/software machine that can ensure satisfaction of the requirements in the problem world,
- The construction of adequacy arguments, convincing both to developers and to customers, users and other interested parties, that the system will provide what is needed.

The application of POSE to Software Engineering design has been demonstrated using case study based research papers and example of which is the use of the framework to solve a package router problem (Rapanotti et al., 2006).

The basic elements of POSE are summarised below:

- Software problems are represented as sequents consisting of a real world context, a requirement and a solution
- POSE uses a stepwise process to solve the software problem. Each step, referred to as a transformation, relates a problem and its justification to its derivative set of problems.
- Problem Transformation Schemas contain information on how a problem is transformed from its premise problem(s) to the conclusion problem.
- Justifications detailing the suitability of a solution for the problem are detailed during problem transformations. The justifications recorded at each transformation step are combined into an “Adequacy Argument” for the solution.

Guarding transformations by justification obligations steps are meant to ensure the problem will be solved in a way that satisfies relevant stakeholders. The adequacy of the application and reasoning behind the decisions reached are recorded by discharging those obligations in the problem transformation steps.

Stakeholder needs will factor in the use of POSE for development (Rapanotti et al., 2006; Hall et al., 2007). Justifications associated with transformations are written in natural language and collectively represent an adequacy argument to provide the basis
for the validating the solution. The criteria may include specific stakeholder requirements.

The application of the POSE framework for the development of a safety critical product with simultaneous production of its safety case is the basis for research work carried out by Hall et al. (2007a). The safety case is documentation providing compelling evidence that the system is adequately safe for a given application in a given environment and of good practice followed during development. A POSE "process pattern for safety critical systems” was developed and applied.

The POSE process framework aims to provide improved traceability of artefacts and their relations increasing the visibility of assumptions for scrutiny and validation. Traceability, in the context of software engineering, refers to the documentation of relationships between layers of information generated during the lifecycle of a software product (Dick, 2005). Traceability of design provides a view of how requirements are met by a software product and can aid in its understanding for future evolution.

Justifications documented in POSE are crucial to the validation of the designed and developed product. They describe how solutions satisfy the requirements. This closer level of relationship between requirements, justifications (and subsequently adequacy arguments) and solutions can be described as “Rich Traceability” (Dick, 2002).

POSE allows any form of reasoning to be accommodated within a single development since the framework does not prescribe a particular format for justifications (Hall et al., 2007a).
The POSE process map was further generalised for application to non-software engineering use resulting in the generic form of the framework – Problem Oriented Engineering POE (Hall et al., 2007c). The generic process map consists of 2 activities – Partial Candidate Problem Exploration and Partial Candidate Solution Exploration and 2 choice points Partial Candidate Problem validation and Partial Candidate Solution Validation. The activities are expressed as partial candidates to allow problem and solution exploration to be focused on partial or higher levels at the beginning of the process when not all the required information may be known.

Hall and Rapanotti (2008b) further enhanced the map as shown in Figure 1 below listing 4 process roles involved in the exploration and validation activities:

- Problem Finder responsible for exploring the problem (1)
- Problem Owning Stakeholder(s) – the Problem Validator(s) of the problems (2)
- Solution Finder responsible for exploring the solution (3)
- Solution Owning Stakeholder(s) - Solution Validator(s) of the solutions (4)
Figure 1 POE Process Pattern

This generalised map was used to demonstrate the capture of knowledge about the publishing industry using POE problem transformation steps (Hall and Rapanotti, 2008a).

The capture of expert knowledge during problem transformations is of particular interest to my research. Zdrahal (2007) presents design as a problem that is ill-structured i.e. one for which the desirable outcome is not known in advance. In this situation, a designer’s domain knowledge facilitates the structuring of design tasks. The design process typically starts with problem analysis which results in a detailed specification. Decisions that are made during the process are based on the designer’s formal and experiential knowledge (Zdrahal, 2007). This knowledge used to reach the decisions and which directs the design towards particular domain(s) is then available for the next stages in the process. However without deliberate actions the domain
knowledge remains with the expert making the decisions (Hu et al., 2000; Tang et al., 2006)

According to Hall and Rapanotti (2008b) Application of the POE process will only result in a problem being solved (i.e., the open circle in Figure 1 reached) when, after one or more iterations, a validated problem is provided with a validated solution. Further complexity of the process map is required to reach the solution achieved by using the POE process map is one of 3 forms, sequentially, in parallel and recursively until the final solution is reached. In sequential application, the end point of a problem cycle is identified as the beginning of another cycle. Many problem solvers can solve problems in parallel with application in the parallel form. The parallel problems may be discovered in a prior sequential application of the problem cycle. The final form of application – recursive or as described by Hall and Rapanotti (2008b) “Fractal” design. A fractal has “self similarity” as a property where splitting a whole into parts results in reduced size copies of the whole (Wikipedia, 2008). This principle is demonstrated in the POE process map when used recursively as the activities of the process – problem and solution exploration, are themselves represented as problems consisting of all the parts of the process map of POE (Hall and Rapanotti, 2008b).

2.1.3 Summary

The aim of this literature survey was to gain an understanding of the work that has been carried out in the areas of design rational capture and problem oriented engineering.
Existing research into design rationale lists additional overheads in terms of cost and time, perceived lack of benefit to practitioners and insufficient tool support as factors that hinder its capture (Tang et al., 2006). Where the rationale is not captured, the reasoning behind designs can often be forgotten or lost when resources move on. Knowledge employed in creating the designs remains tacit amongst “subject matter experts” where an organisation has not explicitly put in place procedures to externalise expert knowledge.

The benefits of recording design rationale are realised by the ability to use the gathered information for further decision making both for the evolution of the designed product and as a basis for the new designs. This makes the elements that are captured and the form in which are represented key factors in design rationale capture.

A lot of the work done on rationale capture and management remains theoretical and recommendations for further research in the area include the reduction in the effort required to document design rationale and increasing its perceived benefit with the intent of increasing the adoption of design rationale capture and management.

Advances in the development of the POE framework which include rich traceability and justification of arguments have led to claims that design rationale is captured as part of the process of applying the framework and that its application is in fact a method of capturing knowledge (Hall and Rapanotti, 2008a).

Table 1 has been drawn up to summarise the common elements of information that are captured to represent design rationale as expressed in reviewed literature.
Organisations are becoming increasingly more dependent on software for carrying out business activities in order to remain profitable. Expertise and knowledge gained in producing software has a significant impact on the organisations’ ability to evolve their
software quickly to remain competitive. Globalisation of the software development activity has created a further need for knowledge capture and transfer to be addressed.

Within the software engineering field, design rationale capture remains largely theoretical. Perceptions appear to have a significant impact on the uptake. These include:

- That the activity is another form of documentation
- That will add further burden to the development activity
- That costs (time and effort) are increased
- That the benefits of the activity are limited

From the foregoing, the following aspects will also be examined as part of the research

- Costs including time and effort
- Benefits including availability of rationale for reuse assistance for decision making and software evolution

2.2 Intended Contribution to Research

This research intends to examine the capture of design rationale during the design process using Problem Oriented Engineering. The generalised POE process map has been used outside of field of software engineering to demonstrate its use specifically for capturing knowledge (Hall et al., 2008a). The knowledge captured during the application of POE to solve a software problem will be assessed against the elements identified in my literature survey to establish if they represent design rationale.
Existing research indicates that the benefits of capturing design rationale are obtained by the ability to access and reuse the information. I intend to hypothesize how this could be achieved by examining the structure of information captured in the use of the POE framework. Designing an automated DR system based on POE is beyond the scope of this research but it is anticipated that information generated from the assessment can be used to recommend extensions to the POE framework for design rationale capture where required and propose future research directions to achieve them.

In summary, this research aims to demonstrate that:

- The information gathered during the case study can be characterized as design rationale
- The structure of the information gathered can facilitate access for later reuse
- The use of POE framework could affect costs and benefits of the design rationale capture activities.
Chapter 3  Research Methods

This chapter outlines the research methods considered and describes the adopted methods and how they support the research aims and objectives.

3.1 Research methods considered

The main aim of this research is investigate the use of the POE framework for the capture of design rationale and whether that rationale can be accessed for reuse. The following research methods were considered for use in the research.

3.1.1 Prototype and Test

Prototyping is a means of demonstrating that a concept can be made to work (M801, 2008). In the context of this research, this would entail creating a prototype of a Design rationale system based on the POE approach for testing. The requirements of the system would have to be established in a form to guide the development of the prototype coded with a suitable development tool. The concept of using POE for design rationale capture is at a very early stage. Having assessed the additional effort required to acquire additional skills necessary to build the prototype this method has been discarded due to the high risk that the objectives could not be achieved in the time allocated for this research project.

3.1.2 Document Studies

This research method was considered and discarded as a main methodology due to the fact that while there is extensive literature available on Design rationale and its capture, the concept of using POE for this purpose is a relatively new concept and
there is insufficient literature to produce a suitably substantial sample of data for research. This would pose a threat to the validity of the research. However the knowledge gained from the literature search formed the basis for evaluating results of the research.

3.1.3 Case Studies

A large proportion of existing research into Problem Oriented Engineering uses the case study approach to describe how the framework can be used. The availability of larger and more realistic case studies from industry is listed as required future research for POE in the software engineering context (Hall et al., 2007c).

3.2 The Selected Approach – Single Case Study

I selected the case study methodology as the main research approach for this research. This method was considered most suitable for the following reasons.

Yin (2003, pg 13) defines a case study as

“An empirical inquiry that investigates a contemporary phenomenon within its real life context especially when the boundaries between the phenomenon and context are not clearly evident”

The phenomenon being investigated is the capture of design rationale and the real life context is collection during the activities to handle an actual software problem in an organisation.

Case studies are suitable in research that involves descriptive evaluation (Yin, 2003 pg 15). This research will analyse POE’s ability to capture design rationale through the
description of its use to solve a software problem. The information collected during the process will then be analysed against factors gathered during my literature survey to determine whether it can be characterised as design rationale.

Opinions and perceptions about design rationale within the organisation were explored using a questionnaire (M801, 2008).

### 3.2.1 Case Study Design

A single case study was used to examine the following:

- How the information gathered during the case study can be characterized as design rationale
- Why the structure of the information gathered can facilitate access for later reuse
- How the use of POE framework could affect costs and benefits of the design rationale capture activities.

Thus there are two main units of analysis – the information gathered and its potential for reuse. Yin (2003) defines this type of case study as a single embedded case study. The rationale for choosing this design is that it will be representative of activities carried out in a typical organisation to resolve a software problem (Yin, 2003 pg 40-41).

Information was gathered by applying POE process steps to transform a problem into a solution acceptable to pertinent stakeholders. The steps were documented using notation prescribed in existing POE literature and the results examined for design rationale elements identified in Table 1 in Chapter 2.
3.3 Threats to validity and ethical considerations

The validity of generalisations obtained from case studies is a common criticism of case study research. Yin (2003, p38) recommends that the research should be generalised to “theory” rather than the specific case. Inductive reasoning was used to make generalisations from the execution of the case study (M801, 2008).

Another method recommended by Yin (1994) to increase the ability to generalise from findings of case study research is the use of triangulation of evidence by taking information from different sources to check validity and reliability of the findings. Information is taken from participant observation, documentation and interviews for this research.

The use of a single case study could be considered a limiting factor on the ability to generalise from the research however, Whyte’s ‘Street Corner Society’ is an example of single case study which is commonly cited due to the ability to generalise from it Yin (1994, p4).

Interactions with participants in the case study organisation were carried out with due care for their general safety. Participants consent was sought to ensure that any time they spent was suitably accounted for. Where required their supervisors’ approval was obtained. The personal details of any individuals contacted for this research have only been used for the purpose described in accordance with legislation set out by the Data Protection Act of 1998.

The company where the case study was based is bound by its parent American financial institution’s strict guidelines on data usage and dispersal. These include regulations imposed by Sarbanes Oxley legislation. I acquainted myself with these
and corporate data usage restrictions to ensure that only information that is permitted is used for this research. Any restricted information has been cleansed.

Extra care was taken to ensure that the confidentiality of the client organisation is maintained.

I ensured that my skills for conducting the research were at a suitable level of competency by consulting literature on my chosen research methodology and believe the case study approach is appropriate for collecting data of suitable quality to answer my research questions.
Chapter 4  Data Collection

4.1 Introduction

This chapter describes how the data was collected for the case study. Terms and notations used in documenting the case are described. The use of a questionnaire to gather information on perceptions of design rationale and the documents that were examined are described. The problem investigated for the case study and the flow of problem information within the organisation are outlined. The POE process roles and steps used in the case study and samples of the diagrams and documentation generated are presented. Finally an outline of analysis used to obtain results is given.

Table 2 is an explanation of terms used in documenting the study.

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client</td>
<td>Refers to the mortgage institution managing customer mortgages</td>
</tr>
<tr>
<td>Financial Institution</td>
<td>Also the client</td>
</tr>
<tr>
<td>Customers</td>
<td>Patrons of the client organisation whose mortgages are being managed</td>
</tr>
<tr>
<td>Mortgage Servicing</td>
<td>Managing mortgage loans – interest accruals, billing, collecting due payments, redemption of loans etc.</td>
</tr>
<tr>
<td>FSA (Financial Services Authority)</td>
<td>Regulatory body for financial institutions</td>
</tr>
<tr>
<td>Mortgage Servicing Software</td>
<td>Software used in servicing activities.</td>
</tr>
</tbody>
</table>
Term | Description
--- | ---
Software supplier | Company that supplies and maintains mortgage servicing software for the client. The case study is based in this organisation
Production Support Teams | Groups of individual tasked with dealing with issues raised in regards to application software being used in day to day business activities
Triage Document | A form used to report details of a production issue
Problem Specification | A document created for a reported issue recording the agreed problem description, analysis carried out, the proposed solution and time estimates

Table 2 Description of Terms

4.1.1 Scope of the Study

The case study was a real development in a Software company solving an issue raised by a one of its clients. The case study was abridged to exclude aspects of the problem which involved information classified as the intellectual property of the Software Supplier. The essential detail and complexities of the design activities carried out is retained to support the aims of my research.

No attempt has been made to prove the use of the POE framework for design or knowledge capture. My literature survey included several documented case studies demonstrating the use of POE in software engineering (Hall et al, 2006, Hall and Rapanotti, 2007, Hall et al, 2007); general engineering (Hall and Rapanotti, 2008a) and for capturing knowledge (Hall and Rapanotti, 2008a).
4.2 The Questionnaire

I decided to use a questionnaire to assess perceptions about design rationale within the company. This method was chosen over face to face interview as it was considered the best means of sampling opinions of staff members globally dispersed over 3 continents. Closed questions using Likert style scales were used. Space was also given for personal comments to be expressed.

The questions were aimed at gathering information of respondents experience and work functions and assessing opinions access to design rationale information affected their work.

The questionnaire was sent out by email to international employees and by hand to UK staff. A covering letter was included requesting assistance a definition of design rationale and instructions for filling the questionnaire. Of the 15 copies distributed, 9 responses were received.

All the respondents had at least 2 years experience in IT and 2 had over 10 years. The respondents covered the following job functions:

- Development
- Support
- Business and systems analysis
- Product assurance (Testing)
- Project and Senior management

The questionnaire responses are discussed further in Chapter 5.
4.3 Document Sources

Documents used as sources of information included:

- Research papers from Literature search: Used to identify elements that represent design rationale
- Triage Document: This document, used to report issues to the Software Supplier, was the source for the initial problem description
- Problem Specification: Contained the validated problem description, clarifications, proposed solution and estimates and was the sign-off document for the solution development commenced.

Samples templates for the Triage and Problem Specification documents are included in Appendix B

4.4 Defining the Valid Amounts Case Study

4.4.1 Overview

The selected case study is set in a UK based subsidiary of an American Financial organisation with clients; business, systems and technical analysts based in the UK; technical architects in the US and development staff in India.

The company recently lost some subject matter experts but still have to provide contractual support to its Client’s to enhance and maintain their supplied software stack.

One of the applications under licence to the Client is the Mortgage Servicing Software package which is used to manage loan accounts once mortgage payouts have been made customers. The software facilitates business tasks such as payment processing and calculations, account queries, early redemptions, correspondence, and interest rate changes and customer billing.
The Software Supplier provides support and assists in the resolution of issues that arise during the use of the supplied software.

Generation of customer billing amounts is based on the terms of the loan, payments received and interest rate changes amongst other factors. Discrepancies in the figures used for calculation can lead to incorrect amounts being billed and/or held as balances against a loan account.

As a financial institution regulated by the FSA, the Client must be able to prove the reasoning behind their charges and services and also provide their customers with enough information on these factors to remain compliant.

The Valid Amounts Problem is based on an issue where the Client identified instances where financial figures used to calculate loan balances and billed amounts were incorrect. Some possible causes of the issue were also identified. A common symptom of the problem was a negative amount on one of the account balances. The problem was given a high priority due to the impact to customers and compliance implications.

The issue was felt to lie with the servicing software processing customer data and was reported to the Software Supplier for investigation and resolution.

The Valid Amounts Problem was chosen as my case study because

- As a member of the Supplier Production Support Team, I had adequate access to the background information, documentation and stakeholders involved in the issue.
- The problem would need a resolution that involved design activity and therefore opportunity to capture design rationale.
The Valid Amount Problem was a typical example of how issues are investigated and resolved in the organisation and was therefore was a suitable candidate as a representative single case study (Yin, 2003 p.41).

### 4.4.2 Handling Problems in the organisation

This section describes the process for handling issues reported to the Software Supplier. Relevant points of flow (labelled a – d in Figure 2) are outlined below.

The initial report may be received via email or telephone call.

**Figure 2 Flow of issue information**

- a) When an issue is found on one of the Software Supplier’s applications being used by the Client, a Triage Document is raised describing the problem and including information that may assist in tracking down its cause. The reported
issue is given a priority by the Client (low, medium, high) which governs the timelines for responses and solutions based on agreed service level agreements (SLAs). The document is sent to the Supplier’s Production Support Team. Once the Triage Document is received, an incident number is generated and used to track the issue. The information is checked to see if it is sufficient for the investigation to progress.

b) Further discussions may be held between the Client and Supplier production support teams to agree that the issue lies with the application software and then an approach for dealing with the issue. Additional clarification may be sought from the business unit where the issue report originated. The requested information may be screen shots of the application error, data extracts, event logs and example scenarios.

c) The issue is analysed and solution approaches proposed by the development and architecture resources assigned to the issue. The proposed solutions are discussed with the Client Production Support Team. Once agreement is reached on a solution approach, it is developed and tested.

d) On completion of development and testing, the solution is packaged with release notes and a test report and delivered to the Client by the Supplier Production Support Team.

e) The Client Production Support Team validate the delivered package, perform some further tests and may either return it for rework if it is unsatisfactory or implement it to the production systems if satisfied with the results.
4.5 Applying POE to the Valid Amounts Problem

In this section, the process applied and results obtained from the use of the POE are documented. The 2 activities – Problem and Solution Exploration, and choice points Problem and Solution Validation of the generic process map (Figure 1 in Chapter 2) were followed to find a solution for the Valid Amounts problem.

Using problem exploration to gain a better understanding of the problem followed by validation of the understanding with the relevant problem holding stakeholder. Solution exploration steps decomposed the problem to a proposed solution architecture which was then validated to ensure satisfaction of requirements and concerns. The process was repeated where necessary to reach a suitable solution.

The design tree in Figure 3 below summarises the steps followed for the case study. I have used a diagram based on an example used to summarise POE analysis in Hall and Rapanotti (2008a). Circles (labelled $P_n$) represent the problem steps, the arrows are transformations labelled $(J_n)$ to show justifications discharged for the steps. I have extended the diagram to show where validation was carried out during the analysis.
I decided to use graphical symbols adopted by Hall and Rapanotti (2008a) to represent problems. Graphical symbols for transformation steps and validations described in Rapanotti and Hall (2008) were also used. The general Justification template described in Hall and Rapanotti, (2008a) was used to document justifications.

Figure 3 Design tree for the Valid Amounts Problem

4.5.1 POE Notation and Diagrams
The scientific notations in existing literature (Hall et al., 2006; Hall and Rapanotti, 2007a; Hall et al., 2007) would be inappropriate in the business setting of the case study.

A summary of the graphical symbols is given in Error! Reference source not found. below. A more detailed explanation of the notations and a full description of the justification template can be found in Appendix B Justification Template.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Context Domain" /></td>
<td>Context Domain</td>
</tr>
<tr>
<td><img src="image" alt="Solution to be found" /></td>
<td>Solution to be found</td>
</tr>
<tr>
<td><img src="image" alt="The Requirement" /></td>
<td>The Requirement</td>
</tr>
<tr>
<td><img src="image" alt="Risk indicator – coded in Red, Amber or Green" /></td>
<td>Risk indicator – coded in Red, Amber or Green</td>
</tr>
<tr>
<td><img src="image" alt="Validation Problem" /></td>
<td>Validation Problem (drawn on the left for Problem Validation and on the right for Solution validation)</td>
</tr>
</tbody>
</table>

**Table 3 POE Diagram Symbols**
4.5.2 Abbreviations

Some abbreviations were used in documenting POE diagrams and justification descriptions. These are described in Table 4.

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPST</td>
<td>Supplier Production Support team</td>
</tr>
<tr>
<td>CPST</td>
<td>Client Production Support team</td>
</tr>
<tr>
<td>PA</td>
<td>Product Assurance (Testing)</td>
</tr>
<tr>
<td>SS</td>
<td>Software Supplier</td>
</tr>
<tr>
<td>DEV</td>
<td>Development Team</td>
</tr>
</tbody>
</table>

Table 4 Abbreviations

4.5.3 The POE Role Assignments

The generic POE process map (see Figure 1 in Chapter 2) list 4 roles – Problem Validator, Problem Finder, Solution Validator and Solution Finder. Table 5 below describes how these roles have been assigned for the Valid Amounts Problem.
## Table 5 POE Role Assignments

<table>
<thead>
<tr>
<th>Roles</th>
<th>Assignment/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem Validator(s)</td>
<td>The Client (the Financial institution managing the mortgages) Main contact will be maintained with the client through the Client Production Support Team (CPST) who manage issues with their production system. The team are part of the IT function and are a group of technically aware individuals who understand the workings of the application system and the platform that is runs on. They provide the initial information on issues and communicate with business decision makers within the organisation when questions arise of a business policy nature.</td>
</tr>
</tbody>
</table>
| Solution Validator(s)        | Three stakeholders fit this role within the Servicing Software company.  
- The UK Supplier production support team –  
  o 1st point of contact with the client  
  o Assign work to the development  
  o Manage releases of solutions to the client  
- The application architect – mainly at the point of review the solution to assess if the solution complies with standards  
- The product assurance team – involved in ensuring the quality of the provided solution  
There are also 3 stakeholders who are the Client’s representatives and have the final say on whether the solution is acceptable  
  - Client Production Support Team  
  - Mortgage business managers within the organisation make the final sign-off decision  
  - IT management make the decision on whether the solution will be implemented to production |
| Problem Finder               | This role will be played by both the Supplier and Client Production support teams                                                                                                                                       |
| Solution Finder              | This will be done by the Supplier Production support team and the Off-shore Development team                                                                                                                                 |

Design Rationale Capture with Problem Oriented Engineering: An investigation into the use of the POE framework for the capture of design and architectural knowledge for reuse within an organisation.
4.5.4 Problem Exploration

The purpose of problem exploration is to capture details of the problem in its defined context and to prescribed requirements. The initial problem details were provided by the Client Production Support Team (CPST) in the form of a triage document describing the problem and other useful information. As a member of the Supplier Production Support Team (SPST), I checked details in the Triage Document to establish they were adequate to facilitate further investigation. Additional examples of the issue and clarifications were obtained from the client.

I used problem exploration on to examine the information on the triage document by the SPST. Using domain interpretation transformation I arrived at the illustration of the Valid Amounts problem shown in Figure 4 below.

![Figure 4 - The Valid Amounts Problem]
There is an explicit requirement to discharge justification obligations for each POE step. The domain interpretation step of the Problem Interpretation transformation schemata class (Hall et al. 2007) was applied to the information in the Triage Document. Descriptions and phenomena of the problem and evidence of their validity are included in the Justification.

**STEP ID: Application of *Domain Interpretation to problem* \( P_0 \)**

**JUSTIFICATION** \( J_1 \): An initial Characterisation of the Valid Amounts Problem

The behaviour of interest is the calculation of amounts to be billed to customers.

The servicing software uses account balances to calculate due payments and pay-off balances for loan accounts. The Client has identified that some balances on loan accounts are incorrect. Where the balances are incorrect, the calculated amounts may also be affected.

**DESCRIPTIONS & PHENOMENA:**

**Descriptions**

<table>
<thead>
<tr>
<th>Accurate Billing</th>
<th>Requirement that customers are correctly billed for the mortgage loans that they have with the Client.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valid Amounts</td>
<td>The solution that is needed to ensure that customers are accurately billed.</td>
</tr>
<tr>
<td>Client</td>
<td>The Financial institution managing customer mortgages</td>
</tr>
</tbody>
</table>
Customers | Customer whose mortgages are being serviced
---|---
Servicing Software | Software used for servicing mortgage loans

### Phenomena

<table>
<thead>
<tr>
<th>Calculations</th>
<th>Application of interest and other factors to produce a figure that the customer should be billed for their loan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bill Amounts</td>
<td>The amounts the customer is advised to pay for a named period (usually monthly)</td>
</tr>
<tr>
<td>Balances</td>
<td>Total amounts for each mortgage loan</td>
</tr>
<tr>
<td>Formulae</td>
<td>Formulae used to calculate due amounts</td>
</tr>
</tbody>
</table>

CONCERN: Validity

STATUS: Pending

A validity concern is raised but remained in a “pending” state until the results of the exploration step were validated.

I observed that the problem exploration step could itself be characterised as a problem within the main POE process having a requirement, solution and context. This smaller internal iteration of the process is described as a “Fractal” POE process pattern in Hall and Rapanotti, 2008b. Using a fractal problem exploration step, the validation criteria for the Valid Amounts problem was obtained from the Triage Document, clarification...
emails and face to face meetings with the Client. The Triage Document was reviewed by the SPST. Where necessary gaps in the provided information are dealt with through follow up

Figure 5 below shows how Problem exploration of the Valid Amounts problem can be represented as a problem.

![Diagram of Partial Candidate Problem Exploration Step as a Problem]

**Figure 5 Partial Candidate Problem Exploration Step as a Problem**

### 4.5.5 Problem Validation

Problem validation is necessary to ensure the problem is understood correctly by both Problem Finder and Validator. The consequences of not validating the problem description could be a solution that does not satisfy the Client’s requirements. Impacts to confidence (that the Software Supplier can provide adequate solutions) and cost (time and effort wasted designing, developing, testing and delivering the wrong
Design Rationale Capture with Problem Oriented Engineering: An investigation into the use of the POE framework for the capture of design and architectural knowledge for reuse within an organisation.

solution) could result if the Solution Finder works on the wrong problem. Visibility of the problem validation requirements and description to both client and supplier ensures that both are aware of what is to be solved by the exercise.

The problem validation step dealt with the need of the SPST, in the role of Problem Finder, to ensure that sufficient detail was available for further analysis of the issue. Figure 6, shows the partial candidate validation problem with the requirement to obtain a valid initial problem description. The stakeholder (Problem Validator) was the SPST and the solution is the adequately populated Triage document. The symbol represents the level of risk – green, amber, red. This convention was used to represent validation steps in the case study.

![Diagram of Problem Validation Triage Document]

*Figure 6 Problem Validation Triage Document*
The risk in this step is shown in red as there is a high risk of the problem not being correctly understood if the triage is unclear or inadequately detailed. This risk must be dealt with before progressing with issue resolution.

**STEP ID: Acceptance of Problem Validation of problem $P_0$**

**CONCERN:** Validity

**STATUS:** Discharged

**CLAIM:** These details provide a valid initial characterisation of the Valid Calculation Amount problem

**ARGUMENT & EVIDENCE:** The information provided on the triage document was examined by the SPST and additional clarifications supplied by the CPST as requested. The information was assessed and deemed sufficient for further investigation. The context interpretation adequately represented the environment the information as reported in the Triage Document by the Client.

**RISK:**
- Insufficient information provided for problem solving
Using context and requirement interpretation led to the Valid Amounts domain being expanded as shown in Figure 7 below into the Mortgage Processing (Calculations) and Loan Balances Domains. Both domains require further investigation.

**Figure 7 - Problem P₁ Expanded Valid Amounts Domain**

**STEP ID: Application of Context and Requirement Interpretation to problem P₁**

**JUSTIFICATION J₂**: Calculations are performed on balance data held for loan accounts to produce billing data for customers. The calculations are defined based on business rules and processes of the client company to produce formulae which are applied to the balances and amounts held for the accounts.
Design Rationale Capture with Problem Oriented Engineering: An investigation into the use of the POE framework for the capture of design and architectural knowledge for reuse within an organisation.

**DESCRIPTIONS & PHENOMENA:**

**Descriptions**

<table>
<thead>
<tr>
<th>Mortgage Processing (Calculations)</th>
<th>The processing, rules and formulae applied to the loan balances on a loan account</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loan Balances</td>
<td>Amounts held for a loan account that are used to carry out calculations – principle balance, interest rate, arrears amounts, prepaid amounts etc</td>
</tr>
</tbody>
</table>

**Phenomena**

<table>
<thead>
<tr>
<th>Calculations</th>
<th>Application of interest and other factors to produce a figure that the customer should be billed for their loan and adjust the balances held for the customer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bill Amounts</td>
<td>The amounts the customer is advised to pay</td>
</tr>
<tr>
<td>Formulae</td>
<td>The methods and calculations applied to generate billing amounts and update account balances</td>
</tr>
<tr>
<td>Balances</td>
<td>Financial figures relating to loan accounts</td>
</tr>
</tbody>
</table>

**CONCERN:** Interpretation Validity

**STATUS:** Discharged

**CLAIM:** The interpretations are valid.

**ARGUMENT & EVIDENCE:** The Valid Amounts domain represents the data held for accounts and the rules/processes applied to that data in order to manage them. The
invalid billing amounts being charged to the customer result from either an issue within the Mortgage Processing (Calculations) domain or within the Loan Balances domain. The requirement to calculate correct billing amounts preserves the original requirement for accurate billing.

Although servicing of accounts covers a vast amount of business processes, for the purposes of this case study, the scope has been limited the show only areas relevant to the problem being examined and therefore represents a valid representation of the problem.

The solution must result in valid amounts being billed to customers. This is necessary for the Client organisation to ensure that

- Customers remain satisfied.
- Compliance requirements of the industry ombudsman, in this case the FSA, are met.

**RISKS**

If the requirement is not met

- Customers continue to be charged incorrectly and are likely to become dissatisfied.
- Customers will not be charged in line with the terms that they have been told leading to compliance issues for the Client.

With a validated view of the problem, focus turned to outlining possible solutions. Figure 8 below shows the initial problem $P_0$ (below the line) transformed following
problem exploration to the expanded problem $P_1$ above the line. The validation step is drawn on the left side to represent the fact that it is a partial candidate problem validation.

Figure 8 Problem transformation – $P_1$ to $P_2$ with Validation
4.5.6 Solution Exploration

Applying solution exploration to problem \( P_2 \) resulted in two sub problems based on the observation that the Valid Amount problem could be solved by either

1) Addressing which lies in the Mortgage Processing (Calculations) domain \( (P_{2,1}) \) or

2) Addressing the resulting symptoms which affect the Loan balances domain \( (P_{2,2}) \)

Looking at the two parts of the problem –

Calculations (in the Mortgage Processing domain)

- If any formula is not correct for particular scenarios (business rules, conditions) then the result will be invalid amounts for billing

Balances (in the Loan Balances domain)

- If any of the balances held on the Loan Account are incorrect then even if the correct calculations are applied, the resulting amounts for billing may be incorrect.

In reality, the two domains were not mutually exclusive as fixing the calculations may not result in the balances being corrected and fixing the balances may not prevent reoccurrence of the issue. Separating the problem in this way allowed for the root cause (entailing application code fix) and effect (entailing data fix) of the issue to be dealt with individually if required and was in line with normal issue resolution practices.

The issue of compliance with FSA regulations was raised by the Client. Customers are informed about the terms of their loans and the advised terms are to be observed in
managing the account. It must also be possible to provide audit trails where changes have had to be made.

I extended the solution justifications to include a compliance concern which then formed part of the validation criteria. The meant that the concern would have to be addressed to the Client’s satisfaction for any proposed solution to be deemed suitable.

The two sub-problems were explored further.

### 4.5.6.1 The Mortgage Processing (Calculations) Sub-Problem

Exploring the Mortgage processing (Calculations) Domain resulted in two components – Interactive (carried out by a user via screen entry) and Batch (bulk handling of accounts outside of the online day) processes which can be used to change loan accounts. Figure 9 overleaf shows the architecture.

In the provided triage information, some functional areas were identified for further investigation as possible root causes of the problem.

In the Interactive Domain

- Manual adjustments
- Redemption activities

In the Batch Domain

- Billing

This was not an exhaustive list of possible root causes.
A solution within this domain would eliminate reoccurrences of the issue and lead to a better application in the long term. It was considered to be a “strategic” approach to solving the issue. The POE justification is documented in Appendix C - Justification for the Mortgage Processing (Calculations) Sub problem.

4.5.6.2 The Loan Balances Sub-Problem

The Loan Balances domain solution would address the balance data held on accounts using a fix program to minimise the impact to customers in the short term. The architecture consisted of two components. The Expected Values domain encompassed the correct balance amounts provided by the Client’s Spreadsheet and the Adjustments...
domain the functionality to apply differences between the actual and expected values to correct the balances.

The incorrect amounts would be reset using figures specified by the Client’s business staff. Figure 10 shows the Loan Balances sub-problem.

![Figure 10 Loan Balances Sub-Problem](image)

The justification for this problem can be found in Appendix C - Justification for the Loan Balances Sub Problem

### 4.5.7 Solution Validation

I assumed the role of application architect in order to validate the two proposed solutions to establish their viability (i.e. that they were within the capabilities of the
company to deliver) and conformance to standards. Details of these validation steps are included in Appendix C.

In the actual handling of the issue, the two solution options were presented for validation by the Client and CPST for discussion.

The Mortgage Processing (Calculations) “strategic” solution although meeting the requirement to produce valid amounts, was rejected at this point by the Client having examined the potential risks.

The justification:

**STEP ID: Application of Solution Validation to problem P2,1**

**Mortgage Processing (Calculations)**

**CONCERN:** Validity

**STATUS:** Un-Dischargeable

**CLAIM:** This solution will meet the Clients requirements

**ARGUMENT & EVIDENCE:**

Main concerns of the Client are compliance and customer satisfaction.

This solution may not resolve the compliance issue without further intervention i.e. the cause is resolved but the effect remains. Customer impact is not addressed in the first instance. There is also a difficulty in determining if all root causes have been
discovered and fixed. The time scales for discovery could prove to be unacceptably long.

**RISKS:**

- Customers impacted for longer
- Amounts may still need to be fixed leaving possibilities of non-compliance with industry regulations
- Some of the root causes are being addressed under other reported issues

The “tactical” Loan Balances solution adequately met requirements as well as addressing compliance and customer satisfaction concerns.

**STEP ID: Application of Solution Validation to problem** \( P_{2,2} \)

Loan Balances

**CONCERN:** Validity

**STATUS:** Discharged

**CLAIM:** This solution will meet the Clients requirements

**ARGUMENT & EVIDENCE:**

Main concerns of the Client are compliance and customer satisfaction.
This solution will resolve the compliance issue by ensuring that balances are correct. The Calculation Spreadsheet provides a documented audit. The solution will also provide a report of the changes made.

Customer impact is addressed in the first instance.

There is still a risk of more customers being impacted since the root causes of the issue are not addressed but as this solution can be applied repeatedly to affected accounts, this risk is considered acceptable.

**RISKS**

- Manual activity required may cause delays
- Root causes will not be addressed

With a solution chosen, the next step would be to progress the solution through detailed analysis. However I observed that no formal vehicle existed to facilitate validation in the processes used by the Software Supplier. In the past, solutions delivered to the Client have been rejected as unacceptable for reasons that could have been established during design. With no formal agreement of the solution in place, the Software Supplier was obliged to rework the solutions.

I held discussions with the Software Supplier Production Support Manager where I gave an overview of the POE process activities and validation check points. I highlighted the absence of a formal sign-off point for an agreed solution as a gap in our internal processes. With management agreement, I devised a document template
called the Problem Specification which included sections for the problem description, clarification details, analysis, initial effort estimates and the proposed solution with examples where possible. This document would serve as a validation and sign-off point, giving the Client an opportunity to review the proposed solution before work was started. My proposal was accepted and the Problem Specification adopted. The new document was accepted by the Client and is now routinely used in dealing with support issues.

The Problem Specification document was used for validation of the Valid Amounts problem. The Validation step is shown in Figure 11 below.

![Figure 11 Valid Amounts Solution Validation](image)

The justification:

The problem descriptions and the selected solution were documented in a Problem specification. This specification was approved by the client for work to progress.
STEP ID: Sign-off of Solution Exploration for P₂

CONCERN: Step Validity

STATUS: Signed-off

DETAILS: The Problem Specification documents validated problem descriptions, clarifications and the proposed solution for the problem as agreed with the Client.

SIGNATORY: Problem Validator - Client

4.5.8 Progressing the Loan Balances Solution

Applying solution interpretation to the Expected Values Domain resulted in the architectural structure in Figure 12. The expected values are produced by the calculation sheet. The sheet applies formulae to specific transactions for a loan to arrive at the values. Justifications for the following sub problems are detailed in Appendix C – Justifications for progressed solutions
The Adjustments domain architecture is shown in Figure 13 and will consist of a program that will use supplied Correction Amounts to fix the invalid balances. Audit reports produced to check that the amounts have been correctly fix and for any future compliance requirements.

Figure 12 Expected Values Architecture

Figure 13 Adjustments
The study was concluded at this point. Figure 14 shows the conclusion of the analysis.

![Figure 14 Expanded Valid Amounts Solution](image)

**Figure 14 Expanded Valid Amounts Solution**

### 4.6 Summary

In this chapter I have described the steps I took to collect data for this research using a questionnaire, documents and by applying the POE process to solve the Valid Amounts Problem case study. I outlined the existing process for receiving and dealing with issues in the Software Supplier. The graphical notation, templates, terms and abbreviations used to document the case study were described. The POE steps applied to solve the problem were explained and documented. Appendix C contains the transformation diagrams and details of the case study not included in this chapter.
Chapter 5  Results

In this chapter, the results of the research are discussed starting an analysis of the data gathered from the questionnaire followed by a review of the information gathered in the case study. The case study data is examined to ascertain if it represents design rationale and the potential for design rationale reuse is considered. The chapter ends with a summary of the research conclusions.

5.1 Results from the Questionnaire

In this section the results of the questionnaire are examined and inductive reasoning is used to make generalizations and casual inferences about the responses received. The number of respondents was considered to provide a suitable representation for use in generalising opinions within the organisation.

Two response ranges were used for the questions –

- Five point scale - Strongly agree (5) – Strongly disagree (1)
- Three point scale – Often (3), Occasionally (2), Never (1)

Responses were plotted on bar charts for illustration.

About design rationale information – What information was considered useful?

Surveyed literature on Design rationale listed its reuse in decision making as a main benefit (Jarczyk et al., 1992). In the software arena, this could relate to software synthesis or evolution. As the case study centred on the support of an existing application, this question focused on software evolution. Respondents were asked to indicate whether they felt certain types of information would help in software
evolution activities. The results (Figure 15) indicated that all respondents agreed or strongly agreed that original requirements for the software and changes to those requirements were useful sources. One could infer from the results that these would be the most commonly used sources of information.

![Figure 15 - Information Helpful for Software Evolution](image)

**Referring to Design Rationale – What sources were used?**

When asked how often certain sources of information were referred to, the responses (illustrated in Figure 16) indicated that original project documentation (within the organisation this would primarily be requirement documents) and subject matter experts were most often used.

Other sources listed on the responses included

- History (of the application)
- Subject matter documents
- Emails between relevant parties
Figure 16 Sources of Information about Software

Accessing Design Knowledge – Why is it not used more?

The third question sought to establish what respondents felt affected their ability to reuse design rationale information. From the results, it is possible to infer that the respondents were most affected by not knowing where to find the information and the loss of information held by individuals. Another factor mentioned was the lack of design documentation for reference. A respondent noted that their function was not usually involved in design and felt that this did cause some issues. In those situations, they usually resorted to business requirement documents.
Capturing Design Rationale – Why doesn’t it happen?

The final question sought to assess what factors were perceived to affect the capture of design rationale in the organisation. The options listed were drawn from my literature survey (Lee, 1997; Tang, 2005). From the results (Figure 18), time constraints appeared to be a major factor in the company. This corresponds to findings in (Tang, 2005). The responses also showed that not knowing what information should be documented affected the recording of rationale. This was also noted in Regli et al. (2000).

Other factors suggested included:

- Type of solution - Strategic or tactical
- Lifetime of the product being designed
- Client motivation
- Lack of comprehensive understanding within the business analysis team of some of the applications; means things can be missed or their significance misunderstood
- Methodology and QA Process
Design Rationale Capture with Problem Oriented Engineering: An investigation into the use of the POE framework for the capture of design and architectural knowledge for reuse within an organisation.

5.2 Case study Results - Design rationale Elements Captured using POE

I examined the information gathered in the case study in light of the Design Rationale elements identified in Table 1 of chapter 2. My findings were that all the identified elements were recorded during the case study leading to the conclusion that Design Rationale is captured when solving a software problem using POE.
In this section I use samples from the case study to describe how the information matches the identified elements. A summary of the results are given in Error! Reference source not found. at the end of the chapter.

5.2.1 Decisions

Design choices are recorded in POE in validation steps. An example of this was observed in the selection of the tactical Loan Balances solution over the more strategic Mortgage Procession option in a solution validation step.

5.2.2 Rationale for Decisions

The generalised POE justification template (Hall and Rapanotti, 2008a) specifies that arguments and evidence be included to substantiate claims about design concerns being met (or not as the case may be). Design choices are guided by the concerns in POE and the arguments and evidence provided form the basis for explaining why those decisions were taken. This was observed in the case study where the need to meet the Compliance concern recorded in justifications was instrumental in choosing the Loan Balances solution (see section 4.5.7).

5.2.3 Alternatives and Trade-offs

Solution exploration results in different potential solutions being identified. There were two possible solutions identified for the Valid Amounts problem. A compromise was made in selecting the Loan Balances solution, electing to leave the root causes to be solved at a later date in favour of satisfying compliance and customer satisfaction concerns quickly.
5.2.4 Suitability

Transformation steps are guarded by justifications in POE and must be “discharged” to establish to stakeholders, the soundness of the resulting problem. Also any proposed solutions must be validated with relevant stakeholders to establish that they are fit for purpose (Hall and Rapanotti, 2008a). Justifications build the adequacy argument for the resulting solution. The Problem Specification Document was used as the basis for formally presenting the proposed solution for validation and acceptance by the Client.

5.2.5 Constraints

Constraints are captured in Concerns, described by Jackson (2001), as something of importance to the development perhaps in relation to a stakeholder. This was observed with Compliance concern which imposed a constraint that had to be satisfied for by the chose solution.

5.2.6 Assumptions

Assumptions are recorded within the descriptions and risks in the justifications for transformation steps. These assumptions are visible for scrutiny and validation to stakeholders. An example of this was observed during execution of the case study (see section 4.5.8) where a risk was registered to note that the Fix Balances solution feasibility rested on the assumption that the supplied fix values were correct.
5.2.7 Status

A status is registered against each concern registered in a justification. These values reflect the position of the transformation step. The step validity concern monitors the status for a particular transformation step.

5.2.8 Complexity of Design

Colwell (2005) describes design complexity as a function of the number of ideas that must be considered simultaneously, the duration that they must be considered and the combination of these two factors. Exploration in POE can focus parts of a problem or at a high level (Hall and Rapanotti, 2007) thus limiting the design ideas that need to be considered in one go but still reflecting the overall complexity of the design when the full transformation from problem to solution is inspected.

5.2.9 Issue Descriptions

Problem exploration is used to arrive at a problem description validated in the accompanying justification. In the case study, the triage document containing the description of the Valid Amounts problem was examined and the resulting problem description validated by the SPST.

5.2.10 Contacts

The four roles in the POE process ensure that relevant stakeholders are identified and involved in the process. Although for the purposes of confidentiality, specific names have not been included for the stakeholders in the case study, they have been identified by their role in the organisations involved (see section 4.5.7, Table 5). In a real life
situation, the names and contact details would be noted alongside their role assignments.

### 5.3 Reusing Captured Design Rationale

The other aspect of interest to this research was whether the design rationale captured with POE could be reused.

I observed that using POE to solve a problem imposed a formal structure on the flow of information used in reaching a solution. Figure 19 shows how the flow of information illustrated in Figure 2 in Chapter 2 looks with the use of the POE process. The diagram has been labelled to highlight points of discussion. Labels a) to e) show how the flow relates to Figure 2. Coloured boxes are used to enclose several steps that refer to the same point in Figure 2 (green for c and red for d). Swim lanes are used to show which POE role has responsibility for each step. Validation points are labelled with a number followed by a single bracket, e.g. 1). Activities are labelled with roman numerals enclosed in square brackets, e.g. [i]. These labels are used to indicate what point in the diagram is being discussed. The blue coloured shapes represent post design activities.

In order to carry out validation there needed to be formal hand-off point between each activity in the case study. For example, the result of initial problem exploration, [ii] on Figure 19, had to be validated so an artefact of some form had to be examined, in the first validation step, this was the Triage Document.
The absence of such a hand-off artefact between the initial problem validation (1) and the Client validating the solution (5), was observed as a gap in the Software Suppliers process. In their normal day to day processes, communication following the acceptance of the Triage Document was transient, centred on emails and face to face discussions. This allowed critical design decisions to be taken about the solution that were not evident to the Client until the solution had been delivered. The lack of a signed-off solution also allowed the Client to introduce changes at late stages of development and reject solutions as unsuitable after delivery.

The introduction of the Problem Specification document (validated at (3)) filled this gap in the case study. The document, designed to include the validated problem description, analysis and clarifications, estimates and the proposed solution created visibility of the solution and related information to relevant stakeholders before significant development work is carried out. The information in the document would also be available to assist decision making should additional work requested in the area. It is noteworthy that after discussions with the Production Support Manager, the document was formally adopted for actual use in the department.

Design rationale reuse could also be available for dealing with alternative solutions for a problem that has already been dealt with. In relation to the case study, the information gathered previously ([i], [ii] & [iii]) would be available for reference, should the Client choose to address root causes of the original problem through the more strategic Mortgage Processing (Calculations) solution. Solution exploration could progress from the point where the Mortgage Processing option was dropped with the previous justifications accessible for consideration.
Design Rationale Capture with Problem Oriented Engineering: An investigation into the use of the POE framework for the capture of design and architectural knowledge for reuse within an organisation.

Figure 19 Flow of Information Revised with the POE process
5.4 Summary

This chapter presented the results of this research which included the outcome of the questionnaire and information from the execution of the case study.

Pattern matching evidence gathered in the case study with the elements of design rationale established from my literature survey, it was possible to show that POE captured information representing each of the Design Rationale elements that had been identified in Table 1. My conclusion from these findings is that design rationale is captured by the application of POE to solve a software problem.

These results are summarised in Error! Reference source not found. below.

<table>
<thead>
<tr>
<th>Information element(s)</th>
<th>In the POE process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decisions</td>
<td>Design choices made Solution validation</td>
</tr>
<tr>
<td>Rationale for decision(s) taken</td>
<td>Reasoning Deliberations Arguments Criteria for selection The argumentation used to support each Justification argument and evidence Concerns</td>
</tr>
<tr>
<td>Alternatives</td>
<td>Trade-offs Solution exploration</td>
</tr>
<tr>
<td>Suitability</td>
<td>Weakness of a design Am I certain that this design would work? Am I certain that I or the team could implement it? Solution validation</td>
</tr>
<tr>
<td>Constraints</td>
<td>Criteria for validation Concerns in justifications</td>
</tr>
<tr>
<td>Assumptions</td>
<td>In justifications</td>
</tr>
</tbody>
</table>
Design Rationale Capture with Problem Oriented Engineering: An investigation into the use of the POE framework for the capture of design and architectural knowledge for reuse within an organisation.

<table>
<thead>
<tr>
<th>Information element(s)</th>
<th>In the POE process</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Status</strong></td>
<td>Tracked in justification concerns</td>
</tr>
</tbody>
</table>
| **Complexity of the design** | POE Diagrams  
To some extent in the justification although not explicitly |
| **Issue descriptions** | Problem exploration  
Problem validation |
| **Traceability**       | Relationships between layers of information  
Transformation steps Diagnos |
| **Contacts**           | Stakeholders  
Represented in POE roles  
Problem finder  
Problem Validator  
Solution finder  
Solution Validator |

*Table 6 Elements of Design Rationale Captured in POE Process Execution*

Also by examining the flow of information (Figure 19), I have been able to demonstrate how reuse of design rationale could be facilitated by the use of the Process. Although the structure of the information was not examined as was the original intent of this research, it was possible to propose ways in which the information could be reused.

Although there is potential for reuse of the rationale, realising that potential will depend on the effectiveness of the categorisation, storage and organisation of the information gathered. In Rapanotti and Hall (2008) where POE was used practically on a project to develop a highly innovative post-graduate research programme for the Open University, categorisation and access were addressed by creating a repository
structured into directories that corresponded to a naming convention adopted for the problems. Thus all the information relating to problem could be accessed by navigating to the directory mirroring its name.

Another important factor observed in this research was that the design rationale was recorded as a methodological by-product (Lee, 1994) of the using the POE process. Using POE for solving software problems with the required validation and justifications carried out for the steps captures the rationale with no additional effort needed. One could conclude from this that the use of POE would assist in addressing cost and time concerns that affect the practice of design rationale capture noted in existing research.

The next chapter summarises the research and suggests areas where further work could be undertaken.
Chapter 6  Conclusions

In this chapter, the research carried out is reviewed and the extent to which objectives were met is examined. The research methods used are assessed for their effectiveness in this research and suggestions for future work are proposed.

6.1 Project review

This research focused on the capture of design rationale while solving a software problem using Problem Oriented Engineering. Research reviewed in the literature survey revealed that whilst the value of capturing design rationale is widely recognised, the practice remains limited due to factors that include cost and time constraints and lack of tools.

Chapter 1 set out the background for the research and outlined the main aims and research question.

Chapter 2 reviewed literature on Design Rationale, its capture and management which yielded a list of elements that would be used as the criteria for identifying design rationale in the information gathered during the case study. Existing research on Problem Oriented Engineering although mostly theoretical, provides evidence of the use of the framework for solving software problems and for capturing knowledge.

The selected research methodology was outlined in Chapter 3. Validity of the chosen methodology and ethical considerations affecting the research were discussed.

Chapter 4 detailed the collection of data from the execution of the case study and the use of a questionnaire aimed at sampling opinions on design rationale and its use in the
study organisation. Chapter 5 analysed the results and outlined conclusions drawn through inductive reasoning.

The following sections review the methodology and aims of the research project.

6.1.1 Methodology

The use of a single case study was the main methodology applied in this research. Although often considered limited in its use for generalisation (Yin, 2003), this methodology was considered suitable as the case study represented a typical case of solving a software problem in an organisation that covered a representative group of stakeholders.

Criteria drawn from existing research was used to establish the existence of design rationale (Table 1). This provided the basis for identifying the existence on rationale in the data gathered and strengthened the validity of the conclusions drawn from the results.

The execution of the case study was carried out for the most part in parallel to the actual problem in the organisation. The case study could have been further strengthened by presenting the results to representatives in the organisation for review.

The case study focused on designing the solution but did not proceed to the details of the coded solution however; I believe the level of information gathered was sufficient to provide the evidence needed.

The questionnaire provided useful insights into opinions and perceptions about design rationale and its management in the organisation. On reflection, I believe that
questions about design rationale elements (i.e. the criteria in Table 1) should have been included, providing a better basis for triangulating the research data.

In terms of the style of questions, selections rather than ranges may have provided more insight to opinions. The questions gave the opportunity for personal comments under the category “Other”. Responses of this type were quite low suggesting that the questions could have been constructed better to elicit open responses.

6.1.2 Achievement of Research Aim and objectives

The main objectives of the research were listed in Section 1.3. These are reproduced below with an assessment of whether they were achieved.

Investigate through a survey of available literature, current research into design rationale and its capture.

There is a wealth of research available for review which provided me with a good understanding of the subject.

Identify from current research and industry guidelines, key data elements that should be captured as design rationale and that facilitate its management and reuse.

The various elements identified in the literature survey were noted and then grouped into a table of information typically recorded as design rationale. This information,
identified in Table 1, served as the criteria for evaluating the data collected during the case study presented in section 5.2 of Chapter 5.

*Investigate the views of typical IT project and support staff on the availability of design rationale in the organisation, its capture and usefulness*

This information was gathered using a questionnaire and sampled views from staff in India and the UK. The results given in 5.1 were used to draw conclusions about perceptions of design rationale within the company.

*Examine, through the use of a case study, which elements of design rationale identified in the literature review are captured by the application of POE to solve a software engineering problem.*

The Valid Amounts problem was explored using the POE process and the data recorded in Chapter 4 and Appendix C. The results of examining the data were presented in Chapter 5 and provided the basis for the conclusion that Design Rationale is represented in the information recorded when the POE process is used to solve a software problem.

*Explore the structure of the data captured by POE to assess how it may be accessible for reuse*

This objective was indirectly achieved by examining the influence of using POE to solve the problem. The intention to examine the structure of the data as stated in this
objective would have taken the research in the direction of synthesising a data repository solution for POE information. I believe that this would have broadened the focus of the research beyond that achievable in the time frame allocated. On reflection, the objective need not have emphasised the structure of the data as a means of determining reuse. Section 5.3 discusses areas where reuse is possible based on the evidence from the case study.

Summarise the results of the research and propose recommendations for future developments and/or further research.

This research has been able to demonstrate that

- The information gathered during the case study can be characterized as design rationale.
- The information gathered can be accessed for later reuse but a strategy for categorising and storing the gathered data will needed for it to be effective.
- The use of POE framework results in the capture of design rationale as the methodological by-product of its use and therefore could affect costs and benefits of the design rationale capture activities.

Recommendations and proposals for future work are presented in section 6.3.

6.2 Discussion

The results of this research, although subject to the limitations of a single case study, have shown that POE could be a valuable methodology for design rationale capture. The benefits to be gained from the practice of rationale capture have been highlighted in research and time and budget constraints on software projects are cited as key factors in restricting its uptake. Having the ability to capture rationale without
additional impact to those factors could encourage greater uptake of the practice. The results of this research have led to the conjecture that POE could be used to accomplish rationale capture through methodological by-product thus helping to address time and budget concerns.

Another area where value was observed in this research was in the management of constraints using concerns expressed in justifications. Since concerns are of relevance to the stakeholders involved in the process that will have a role in validating solutions, it is possible to enforce these constraints in a way that ensures they guide the solution that is developed. This was observed in the case study where the compliance concern was included to cater for the Client’s obligation to meet industry regulations imposed by the FSA. The chosen solution had to satisfy this concern in order to be acceptable and influenced the final solution with the inclusion of audit reports which could be used to meet compliance obligations.

The use of the POE process led to the creation of a new document - the Problem Specification document to satisfy validation requirements imposed by the process. The document has been adopted and is in practical use by the Software Supplier. It closed a gap in their process which allowed solutions that were not suitable to be delivered and subsequently rejected. It also serves as a source of design rationale information as it includes reasoning and clarifications about the chosen solution. The Software Supplier experienced a reduction in the number of solutions returned for rework following the introduction of this document.
6.3 Recommendations for Future Work

Three areas have been identified as possible subjects of further research.

Benefits of Using POE for Capture Design Rationale

The results of this research have led to the conjecture that POE can positively address the time and budget limitations that inhibit design rationale capture. A more detailed investigation into the potential benefits that could be achieved in these areas could have a positive effect in promoting design rationale capture as well as encouraging the uptake of the POE within organisations.

Categorisations, Storage And Access Methods for Information Gathered Using POE

Hu et al., (2000) surmise that the way in which design rationale will be stored and presented for retrieval is of critical importance. This will support the communication and reuse of design knowledge. A large amount of information can be generated from the use of POE and effective categorisation and storage for this information is an area that would benefit from further research. A standard repository, indexed information and access methods for the information are all areas where further research would be beneficial.

Tool Support for POE

Lack of tool support for design rationale capture is also identified as a limitation to its uptake (Tang et al., 2006). Architecture and problem transformation diagrams, validations and justifications are generated during the use of the POE process. These diagrams and documentation have been generated for this research using word processing, presentation and drawing tools. The availability of a single tool that can be
used to generate the required documentation for POE (and perhaps aid in its categorisation and storage) is another area that could be further researched.

This chapter presented a summary of the research undertaken. The degree to which research objectives were met was examined. Valuable observations from the research were discussed and finally suggestions for future research were proposed.
References


Design Rationale Capture with Problem Oriented Engineering: An investigation into the use of the POE framework for the capture of design and architectural knowledge for reuse within an organisation.


Bibliography


Design Rationale Capture with Problem Oriented Engineering: An investigation into the use of the POE framework for the capture of design and architectural knowledge for reuse within an organisation.

## Index

| Apprentice, 13, 24 | Mortgage Servicing Software, 42, 45 |
| argumentation, 14, 34, 86 | phenomena, 27, 55, 114 |
| **Assumptions**, 34, 81, 86 | POE framework. See POE |
| Automatic generation, 13, 24 | POE process map, 32, 35, 52 |
| **Complexity of the design**, 34, 87 | Problem Finder, 30, 52, 57, 58 |
| constraints, ix, 12, 22, 78, 89, 105, 107, 110 | Problem Oriented Engineering, 15, See POE |
| **Constraints**, 34, 81, 86, 127 | Problem Oriented Software engineering - **POSE**, 15 |
| corporate memory, 20 | Problem Owning Stakeholder. See Problem Validator |
| Data Protection Act of 1998., 40 | Problem Specification, 43, 45, 70, 71, 72, 81, 84 |
| descriptions, 14, 27, 34, 52, 71, 72, 81, 87, 114 | Problem Transformation Schemas, 28 |
| Design assumptions. See Assumptions | Problem Validator, 30, 52, 58, 72, 87 |
| Design constraints. See Constraints | process model, 26, 27 |
| Design rationale, ix, 10, 11, 21, 37, 75, 79, 84, 99, 107 | questionnaire, 39, 42, 44, 74, 75, 79, 86, 89, 90, 92 |
| design tree, 49, 115 | **Real World Context.** See Software Problems |
| Financial Services Authority, 42 | Reconstruction, 13, 24 |
| Fractal, 32, 56, 98, 100 | Record and replay, 13 |
| FSA. See Financial Services Authority | Record and Replay, 24 |
| graphical notation, 74 | representation schema, 26, 95 |
| IEEE Recommended Practice for Architectural Description of Software-Intensive Systems. See industry guidelines | Requirement. See Software Problems |
| industry guidelines, 14, 18, 91 | service level agreements, 48 |
| justification obligations. See justifications | Servicing Software. See Mortgage Servicing Software |
| Justifications, 28, 29, 72, 81, 131 | SLA. See Service Level Agreements |
| methodological by-product, 19, 88, 93 | Software problems, 27, 28 |
| Methodological by-product, 13 | Solution. See Software Problems |
| Methodological By-Product, 24, 25 | Solution Exploration, 30, 49, 64, 71, 132, 133 |
| methods of capturing design rationale, 13, 24 | Solution Finder, 30, 52, 58 |
| Solution Owning Stakeholder. See Solution Validator | transformation, 28, 31, 50, 54, 55, 74, 81, 82 |
| Solution Validator, 30, 52, 87 | Triage Document, 43, 45, 47, 54, 55, 56, 59, 83, 84 |
| Stakeholders, 34, 87 | validity, 38, 40, 55, 56, 82, 90, 115 |
| tacit knowledge, 12, 25 | Views and Beyond approach. See industry guidelines |
Appendix A - Questionnaire

Dear Colleague,

Attached is a short survey that I am conducting as part of an Open University postgraduate research project concerning the capture and reuse of design rationale within an organisation.

For the purpose of clarity, I am using the term design rationale to mean the decisions taken, constraints applied and other information that led to the solution that was developed for a requirement (or set of requirements). In other words, why an application is the way it is.

One of the aims of my project is to investigate the factors that affect the capture of design rationale and how it is.

During the course of your experience you are likely to have encountered situations where the lack of background knowledge for an application or function may have had an impact on the work you were trying to complete and I believe our views would be valuable as input to my research.

I would be grateful if you could spare about 15 minutes of your time, to answer the attached questions and return the completed form to me by return email by Monday 10th November 2008.

This is a non commercial project and all the data collected will only be used for my OU project. Please do not supply any confidential information. I will not include any names or confidential information in the reports/publication that results from this survey and completion is voluntary.

If you have any questions please give me a call on the above number

Yours sincerely

Ann Nkwocha
Please base your answers on your own experiences and if you find any question inappropriate to your experience, please leave it blank.

I – About you

a) What is your main work function?

(Please tick as appropriate – you may check more than one box if appropriate to your role)

<table>
<thead>
<tr>
<th>Function</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Analysis</td>
<td></td>
</tr>
<tr>
<td>Systems Analysis</td>
<td></td>
</tr>
<tr>
<td>Programming – New Development</td>
<td></td>
</tr>
<tr>
<td>Programming – Maintenance of existing systems</td>
<td></td>
</tr>
<tr>
<td>Application support</td>
<td></td>
</tr>
<tr>
<td>Project Management</td>
<td></td>
</tr>
<tr>
<td>Senior management</td>
<td></td>
</tr>
<tr>
<td>Other (please specify below)</td>
<td></td>
</tr>
</tbody>
</table>

b) How many year of experience in IT do you have?

<table>
<thead>
<tr>
<th>Years</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Under 2 years</td>
<td></td>
</tr>
<tr>
<td>2 – 5 years</td>
<td></td>
</tr>
<tr>
<td>6 – 10 years</td>
<td></td>
</tr>
</tbody>
</table>
2 - About the availability and reuse of Design rationale in the organisation

a) Having access to the following information would help when trying to investigate a problem for enhancement or support

<table>
<thead>
<tr>
<th>Information</th>
<th>Strongly Agree</th>
<th>Agree</th>
<th>Neither agree nor disagree</th>
<th>Disagree</th>
<th>Strongly disagree</th>
</tr>
</thead>
<tbody>
<tr>
<td>The original requirement for the function</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The constraints that affected the requirement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Changes to the original requirement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A description of the proposed solution to meet the requirement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A description of the solution that was delivered</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decisions made that changed the solution</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The people involved in making the decisions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
b) When you need some background information what sources do you refer to

<table>
<thead>
<tr>
<th>Information source</th>
<th>Often</th>
<th>Occasionally</th>
<th>Never</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject matter expert</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Original project documentation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>End users</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source code</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other (please specify below)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

c) Which of the following factors do you think most affects your ability to reuse of design knowledge?

<table>
<thead>
<tr>
<th>Factor</th>
<th>Often</th>
<th>Occasionally</th>
<th>Never</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessibility of the information – knowing where to find it</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Lack of standard method for representing information

Availability - information lost with individuals who have moved on

Other (please specify below)

d) Please indicate your opinion on the following statements

<table>
<thead>
<tr>
<th>Information</th>
<th>Strongly Agree</th>
<th>Agree</th>
<th>Neither agree nor disagree</th>
<th>Disagree</th>
<th>Strongly disagree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Having access to design rationale will help me in my work</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Having access to design rationale would benefit the organisation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

e) What factors do you think affects the capture of design rationale?

<table>
<thead>
<tr>
<th>Information</th>
<th>Strongly Agree</th>
<th>Agree</th>
<th>Neither agree nor disagree</th>
<th>Disagree</th>
<th>Strongly disagree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem Area</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------------------------------------------------------------------</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lack standard method for recording the information</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time constraints during the project life cycle</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Perceived benefit of the information</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Knowing what information to record</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other (please specify below)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Results (raw)

<table>
<thead>
<tr>
<th><strong>main work Function</strong></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Analysis</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Systems Analysis</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Programming – New Development</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Programming – Maintenance of existing systems</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Application support</td>
<td>x</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Project Management</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Senior management</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Other (QA)</td>
<td></td>
<td></td>
<td>x</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Years of experience</strong></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Under 2 years</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 – 5 years</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>6 – 10 years</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Over 10 years</td>
<td>x</td>
<td></td>
<td>x</td>
</tr>
</tbody>
</table>

**a) Would having access to the following information would help when trying to investigate a problem for enhancement or support?**

(SA=5; AG=4; NA=3; DI=2; SD=1)

<table>
<thead>
<tr>
<th>Information</th>
<th>5</th>
<th>4</th>
<th>5</th>
<th>5</th>
<th>5</th>
<th>5</th>
<th>5</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>The original requirement for the function</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>The constraints that affected the requirement</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>Changes to the original</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>
Design Rationale Capture with Problem Oriented Engineering: An investigation into the use of the POE framework for the capture of design and architectural knowledge for reuse within an organisation.

<table>
<thead>
<tr>
<th>requirement</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>A description of the proposed solution to meet the requirement</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>A description of the solution that was delivered</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Decisions made that changed the solution</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>The people involved in making the decisions</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

b) When you need some background information what sources do you refer to (Often=3; Occasionally=2; Never=1)

<table>
<thead>
<tr>
<th>Source</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject matter expert</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Original project documentation</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>End users</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Source code</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Other (please specify below)</td>
<td>History</td>
<td>System matter docs</td>
<td>emails b/w parties</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

c) Which of the following factors do you think most affects your ability to reuse of design knowledge? (Often=3; Occasionally=2; Never=1)
### Design Rationale Capture with Problem Oriented Engineering: An investigation into the use of the POE framework for the capture of design and architectural knowledge for reuse within an organisation.

<table>
<thead>
<tr>
<th>Accessibility of the information – knowing where to find it</th>
<th>3</th>
<th>3</th>
<th>1</th>
<th>3</th>
<th>3</th>
<th>2</th>
<th>2</th>
<th>3</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lack of standard method for representing information</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Availability - information lost with individuals who have moved on</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Other (please specify below)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

**d) Please indicate your opinion on the following statements (SA=5; AG=4; NA=3; DI=2; SD=1)**

<table>
<thead>
<tr>
<th>Having access to design rationale will help me in my work</th>
<th>4</th>
<th>4</th>
<th>5</th>
<th>5</th>
<th>4</th>
<th>5</th>
<th>5</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Having access to design rationale would benefit the organisation</td>
<td>0</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

**e) What factors do you think affects the capture of design rationale? (SA=5; AG=4; NA=3; DI=2; SD=1)**

<table>
<thead>
<tr>
<th>Lack standard method for recording the information</th>
<th>4</th>
<th>5</th>
<th>4</th>
<th>5</th>
<th>2</th>
<th>4</th>
<th>3</th>
<th>3</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time constraints during the project life cycle</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Perceived benefit of the information</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Knowing what information to record</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>
Appendix B – POE Notation and Templates

Justification Template

**STEP ID: Application of NAME to problem P**

**JUSTIFICATION J:** A justification can be named for ease of reference.

**DESCRIPTIONS & PHENOMENA:** The collection of descriptions and phenomena of the domains and requirements introduced into the problem by the step or the manipulations defined thereon by the step. For an application of the Context Interpretation step, for instance, a detailed description of the elements of $W$ and $W_0$ would be given, alongside any relationship that holds between them, such as shared descriptions, etc.

**CONCERN:** Name

**STATUS:** Status

A concern (Jackson, 2001) is something that is important to the development, presumably because it relates to some stakeholder in the process. In high integrity development, for instance, the reliability concern is likely to arise; a design that does
not address such a concern in such a context is likely to be unvalidatable. The status of a concern is one of pending, discharged, un-dischargeable. The work appertaining to the discharge of a concern is structured: each concern has associated with it the following:

CLAIM: The statement of the claim(s) that will discharge the concern;

ARGUMENT & EVIDENCE: The reason to believe each claim (or the reason it does not hold);

RISKS:

- A description of the risks involved in continuing the development should the concern fail to be discharged, and/or the secondary risk introduced by the discharge of the concern.
- A description of the treatment of risks residual to the step.

A concern established as part of a step may be addressed (and therefore discharged) in design steps subsequent to that in which it is established, i.e., when, as part of other design steps, evidence in support of its associated claim is discovered. The argument and evidence may, therefore, make reference to other concerns, arguments and evidence in the design tree. The validity concern for a step that will, subject to external validation by problem- and solution-owning stake-holders, typically be required to ensure that relationships between concerns and their discharge are adequate.

CONCERN: Step Validity

STATUS: Status

The status of the step validity concern, possible values include pending, signed-off, un-dischargeable
ARGUMENT & EVIDENCE: Explanation of the status after validation, including the relationships here evidence was gathered in the design, and the treatment chosen for the residual risk of the step.

SIGNATORY: To recognise the stake-holder or stakeholders that signed-off the step
Incident Triage Information

- White areas to be completed by report initiator

<table>
<thead>
<tr>
<th>Reference: (Supplied by CW)</th>
<th>Date/Time:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reported By:</td>
<td>Phone:</td>
</tr>
<tr>
<td>Organization:</td>
<td>Email:</td>
</tr>
<tr>
<td>Severity:</td>
<td></td>
</tr>
<tr>
<td>Level 1 ☐ Level 2 ☐ Level 3 ☒ Level 4 ☐</td>
<td></td>
</tr>
<tr>
<td>Status:</td>
<td>Status Date:</td>
</tr>
<tr>
<td>Client Reference:</td>
<td>Client Impact:</td>
</tr>
<tr>
<td>Problem Title:</td>
<td></td>
</tr>
<tr>
<td>Problem Description:</td>
<td></td>
</tr>
<tr>
<td>Applications Affected:</td>
<td></td>
</tr>
<tr>
<td>Other systems Impacted:</td>
<td></td>
</tr>
<tr>
<td>Expected &amp; Actual Results:</td>
<td></td>
</tr>
</tbody>
</table>
Design Rationale Capture with Problem Oriented Engineering: An investigation into the use of the POE framework for the capture of design and architectural knowledge for reuse within an organisation.

### Supporting Documents Attached:

*screen shots etc*

### Steps to Recreate Defect:

#### Resolution Time Line:

#### Solution Short Term:

#### Solution Long Term:

#### Target Release:

### Notes

Include as much detail as possible in this document

### Supporting Documentation

Include any screen shots of the application, error messages, reports, error logs etc that will assist in the diagnosis and resolution of the incident.

### Assigning Severity Levels.

The table below should be used as a guide for assigning the severity of the Defect being reported and the target resolution times

<table>
<thead>
<tr>
<th>Severity</th>
<th>Description</th>
<th>Initial Response</th>
<th>Target to Workaround</th>
<th>Target to Resolution</th>
<th>Update Frequency</th>
<th>Management Escalation</th>
<th>Incident Review &amp; Report</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The production use of the Licensed</td>
<td>30 minutes</td>
<td>&lt; 2 hour</td>
<td>72 hours</td>
<td>60 minutes</td>
<td>30 minutes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

---

Open University – M801 Research Project and Dissertation

Ann Nkwocha

Page 117 of 135
Design Rationale Capture with Problem Oriented Engineering: An investigation into the use of the POE framework for the capture of design and architectural knowledge for reuse within an organisation.

<table>
<thead>
<tr>
<th>Severity</th>
<th>Description</th>
<th>Initial Response</th>
<th>Target to Workaround</th>
<th>Target to Resolution</th>
<th>Update Frequency</th>
<th>Management Escalation</th>
<th>Incident Review &amp; Report</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Software is stopped or severely impacted that Licensee cannot reasonably continue operating the Licensee Business. Licensee experiences a complete loss of service. The operation is mission critical to the Licensee's Business and the situation is an emergency.</td>
<td>30 minutes</td>
<td>&lt; 4 hours</td>
<td>72 hours</td>
<td>60 minutes</td>
<td>60 minutes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

2. Licensee experiences a severe loss of service. Important features are
Design Rationale Capture with Problem Oriented Engineering: An investigation into the use of the POE framework for the capture of design and architectural knowledge for reuse within an organisation.

<table>
<thead>
<tr>
<th>Severity</th>
<th>Description</th>
<th>Initial Response</th>
<th>Target to Workaround</th>
<th>Target to Resolution</th>
<th>Update Frequency</th>
<th>Management Escalation</th>
<th>Incident Review &amp; Report</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.</td>
<td>Licensee experiences a minor loss of service. The impact is an inconvenience, which may require a Workaround to restore functionality.</td>
<td>4 hours</td>
<td>&lt; 8 hours</td>
<td>240 hours</td>
<td>4 hours</td>
<td>48 hours</td>
<td>On request</td>
</tr>
<tr>
<td>4.</td>
<td>Licensee requests information clarification regarding the</td>
<td>8 hours</td>
<td>&lt; 16 hours</td>
<td>240 hours</td>
<td>On request</td>
<td>On demand</td>
<td>No</td>
</tr>
</tbody>
</table>
Design Rationale Capture with Problem Oriented Engineering: An investigation into the use of the POE framework for the capture of design and architectural knowledge for reuse within an organisation.

<table>
<thead>
<tr>
<th>Severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Licensed Software but there is no impact on the operation of the Licensed Software. Licensee experiences no loss of service. The result does not impede the operation of the Licensed Software</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Initial Response</th>
<th>Target to Workaround</th>
<th>Target to Resolution</th>
<th>Update Frequency</th>
<th>Management Escalation</th>
<th>Incident Review &amp; Report</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Sample problem specification

Problem Specification

Title...

Date:

Document Version:

Prepared By:
Introduction

The purpose of this document is to described the reported problem and its features, identify any clarifications and to provide a outline of the proposed solution so that the requirements and scope can be verified.

The Reported Issue

Overview of Reported Issue

Description of the problem that occurred

Impact

Known impact to business process, customer etc

Examples/Known Scenarios

Screenshots, spreadsheets, scenario descriptions provided

Initial Analysis

Description of analysis carried out so far

Issue Recreation

Was it possible to recreate the issue in our environments?
Design Rationale Capture with Problem Oriented Engineering: An investigation into the use of the POE framework for the capture of design and architectural knowledge for reuse within an organisation.

<table>
<thead>
<tr>
<th>Sr No.</th>
<th>Issue Identified</th>
<th>Comment/Information required</th>
<th>Recreated (yes/No)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Clarifications

<table>
<thead>
<tr>
<th>Sr No.</th>
<th>Issue Identified</th>
<th>Comment/Information required</th>
<th>Clarification</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The Proposed Solution

*Fix Requirements*

Describe the proposed solution and list in the table what it will fix

<table>
<thead>
<tr>
<th>Sr No.</th>
<th>Issue Identified</th>
<th>Requirement for solution</th>
<th>Included in this Fix? (Yes/No)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Open University – M801 Research Project and Dissertation

Ann Nkwocha

Page 123 of 135
Initial estimate

An initial guesstimate of the work required as a guide for dates (further updates to be given as elapsed time)

Sign-offs

Add approvers as necessary for this document.

<table>
<thead>
<tr>
<th>Name</th>
<th>Title</th>
<th>Signature</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology Manager</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Technical Specialist</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Client Technical Manager</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Client Technical Specialist</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix C – Case Study Details

Justification for the Mortgage Processing (Calculations) Sub problem

**STEP ID: Application of Solution Interpretation to problem** $P_{2,1}$

**JUSTIFICATION J$_3$:** Calculations are performed on loan accounts to produce billing amounts for customers. Some of root causes identified for this issue had already been raised as separate issues under investigation with the Software Supplier.

**DESCRIPTIONS & PHENOMENA:**

**Descriptions**

<table>
<thead>
<tr>
<th>Batch Processing</th>
<th>Processes applied across the accounts during batch processes.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interactive Processing</td>
<td>Processes applied online but clients agents dealing with customer accounts</td>
</tr>
</tbody>
</table>

**Phenomena**

<table>
<thead>
<tr>
<th>Rules</th>
<th>Business rules that are applied when dealing with client accounts - restrictions and/or compulsory factors that apply to loan accounts. These can affect the calculations/formulae applied.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formulae</td>
<td>Methods to be applied in calculating balances</td>
</tr>
<tr>
<td>Conditions</td>
<td>Special scenarios that a loan may be in – e.g. Arrears on payment holiday</td>
</tr>
<tr>
<td>Transactions</td>
<td>Monetary movements, rate changes, charges, bills</td>
</tr>
</tbody>
</table>
CONCERN: Feasibility

STATUS: Discharged

CLAIM: This is a feasible solution for the problem

ARGUMENT & EVIDENCE:

Identifying commonality between affected accounts could lead to automated resolutions for the problem. Addressing root causes and providing fixes to resolve them stop the problem from increasing. Root causes can occur when

- Formulae applied are either not correct or applied incorrectly
- Constraints imposed by business rules are not applied correctly
- Validations carried out internally or on user interface may be deficient
- Root causes can occur during batch or online (interactive processing)

Advantages of this approach

- Little or no manual intervention required once in place
- The root causes creating the issues will be investigated and eliminated from the system thereby restoring data integrity
- Automation could significantly reduce the amount of time required to eradicate the problem and its effects

RISKS:

- Inability to identify and fix all root causes
- Longer period of impact to customers
- Balances are not fixed by addressing root causes

CONCERN: Compliance
STATUS: Pending

CLAIM: This solution will result in the Client meeting its compliance obligations

ARGUMENT & EVIDENCE:

Resolving the root causes of the issue will prevent further occurrences but may not actually fix the balances that are incorrect without further intervention.

RISKS:

- Balances not fixed by correcting root causes

CONCERN: Customer Satisfaction

STATUS: Pending

CLAIM: This solution will result in the reduced impact to customers

ARGUMENT & EVIDENCE:

While further occurrences of the issues will be prevented, the customers already impacted may still require further intervention.

RISKS:

- Customers may be impacted for longer whilst the solution is being developed
- Amounts may still need correcting after the root cause has been addressed
Justification for the Loan Balances Sub Problem

STEP ID: Application of Solution Interpretation to problem P_2,2

JUSTIFICATION J_4: This solution would correct the affected loan balances using a generic "data fix" program, allowing the client to enter the correct financial figures to be updated against loan accounts. This solution would need to be used repeatedly to fix batches of affected accounts.

Descriptions

<table>
<thead>
<tr>
<th>Expected Values</th>
<th>These are the amounts that the Client believes should apply. They are derived from the Calculation Spreadsheet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adjustments</td>
<td>These changes to be made to the actual balances to produce the expected values. To include an audit report of changes made.</td>
</tr>
</tbody>
</table>

Phenomena

<table>
<thead>
<tr>
<th>Differences</th>
<th>These are between the actual and expected balance values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Validation Rules</td>
<td>Criteria applied to ensure correct amounts are generated</td>
</tr>
</tbody>
</table>

CONCERN: Feasibility

STATUS: Discharged

CLAIM: This is a feasible solution for the problem

ARGUMENT & EVIDENCE:
The main impact of the Valid Amounts problem is to balances on the loan accounts. This solution would correct the amounts removing that impact.

Producing the correct figures to be used and proving how they were calculated is accomplished using a complex spreadsheet template manually populated with the financial figures from the application system. These figures are the checked and the final figures to be used have to be signed off by Business stakeholders before the fix can be applied to production data.

Advantages of this approach

- Allows the client to target and correct most impacted customers
- The generated spreadsheet provides the required proof as required by the regulatory body for the amounts arrived at.
- The quick turnaround means the issue can be addressed with the required urgency

RISKS:

- High level of manual activity
- Possibility of introducing new errors
- The root causes not addressed leading to the number of affected accounts increasing over time
- A long projected timeline to fix all the identified accounts due to the manual effort involved

CONCERN: Compliance

STATUS: Discharged

CLAIM: This solution will result in the Client meeting its compliance obligations
ARGUMENT & EVIDENCE:

Customers may have to be advised of changes to their account to resolve this problem in order to remain compliant. The spreadsheet used to generate the expected values serves as a validation and sign-off artefact for the business and provides proof that can be used to show compliance.

RISKS:

- Balances not fixed by correcting root causes

CONCERN: Customer Satisfaction

STATUS: Discharged

CLAIM: This solution deal with customer impact.

ARGUMENT & EVIDENCE:

This solution removes the impact to customer loan accounts resulting from the Valid Amounts Problem as the balances and billing amounts are corrected.

RISKS:

- High level of manual activity may cause delays
- Root causes are not addressed so customers may be impacted again.

Justifications for progressed solutions
**STEP ID: Application of Solution Exploration to problem P₃**

**JUSTIFICATION J₃**: The amounts (balances) generated by the Clients calculation sheet are to be used as the values for correcting the invalid amounts on the loan accounts.

**Descriptions**

<table>
<thead>
<tr>
<th>Calculation sheet balances</th>
<th>These are Balance amounts generated by the spreadsheet.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loan Transactions</td>
<td>These are the billing, payment and other specified transactions</td>
</tr>
<tr>
<td>Spreadsheet Formulae</td>
<td>The formulae applied to the transactions to provide the expected values</td>
</tr>
</tbody>
</table>

**CONCERN**: Feasibility

**STATUS**: Discharged

**CLAIM**: This is a feasible solution for the problem

**ARGUMENT & EVIDENCE:**

The calculation sheet has been created and validated by the Client. Using the expected values from this source will fix the invalid amounts to Client requirements.

**RISKS:**

- Possibility of introducing new errors through manual activity
- A long projected timeline to fix all the identified accounts due to the manual effort involved
**STEP ID: Application of Solution Exploration to problem P_4**

**JUSTIFICATION J_6:** A fix program is required to amend the invalid balances using correction amounts.

**Descriptions**

<table>
<thead>
<tr>
<th>Correction Amounts</th>
<th>These are the differences between the expected and actual balances that will be applied to the loan accounts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audit</td>
<td>Reports used to validate the results of the fixing the accounts and for any future compliance requirements</td>
</tr>
<tr>
<td>Fix Data</td>
<td>The program to apply the fix amounts</td>
</tr>
</tbody>
</table>

**CONCERN:** Feasibility

**STATUS:** Discharged

**CLAIM:** This is a feasible solution for the problem

**ARGUMENT & EVIDENCE:**

The invalid loan balances will be adjusted to meet Client defined amounts. The changes can be validated and accepted or rejected. Compliance concerns can be addressed with the Audit reports generated by the Fix program.

**RISKS:**

- Based on the assumption that the Fix amounts provided are correct
Design Rationale Capture with Problem Oriented Engineering: An investigation into the use of the POE framework for the capture of design and architectural knowledge for reuse within an organisation.

The Valid Amount Solution
Solution Validation - Viability