Softw Syst Model (2013) 12:825–845
DOI 10.1007/s10270-011-0223-y

REGULAR PAPER

Design rationale capture for process improvement in the globalised enterprise: an industrial study

A. Nkwocha · J. G. Hall · L. Rapanotti

Received: 29 March 2011 / Revised: 14 September 2011 / Accepted: 21 November 2011 / Published online: 15 December 2011

Abstract Design rationale 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 Software Engineering corroborates the importance of design rationale to capture knowledge assets, particularly in the context of the global enterprise, with its increased reliance on IT systems, and risk of knowledge loss through staff movement and attrition. Despite this, the practice of design rationale capture is not as extensive as could be expected due to reasons which include time and budget constraints, the lack of standards and tools, and some uncertainty as to its actual added value. In this paper, we address the viability and benefits of capturing design rationale as a by-product of design in the context of a real-world global organisational setting. This was achieved through a study in which an emerging design approach—Problem Oriented Engineering—was applied in the context of a global financial institution to address a critical IT problem as part of its software supplier's client resolution process. The study provides some positive evidence that the approach-guided knowledge capture of key design rationale elements and that it combined well with existing practices within the organisation and even led to improvement to one of their key processes.

Communicated by Prof. Ketil Stølen.

A. Nkwocha
CW Tech Solutions Ltd, Bank of America, Dartford, UK
e-mail: ann.nkwocha@bankofamerica.com

J. G. Hall · L. Rapanotti (✉)
Department of Computing, The Open University, Milton Keynes, UK
e-mail: L.Rapanotti@open.ac.uk

J. G. Hall
e-mail: J.G.Hall@open.ac.uk

Keywords Design rationale · Process improvement · Problem oriented engineering · Assurance-driven design

1 Introduction

A Gartner report on IT services [3] projected that almost half of all Fortune 1000 global enterprises would choose to draw business benefits from IT service providers, rather than owning their IT assets. Buying into IT service provision may reduce fixed costs—for instance, there will be a reduction in experienced staff needed—and may free an organisation to focus on their core business without day-to-day IT distractions, but there may be a concomitant increase in exposure to risk—for instance, that the system is of adequate quality, or that the provider will be able to sustain any service level agreement in force for the duration of the relationship. With many organisations having rushed to explore the benefits of IT outsourcing, some find themselves moving back to in-house provision.

No matter whether the move is in-to-out or out-to-in, one hurdle in the way of change is the difficulties of sharing and/or relocating knowledge and expertise between two separate organisations. Expertise is achieved as a result of dedicated application within a chosen field, having been exposed to many examples of the problems and solutions that occur therein [13]. It is as expensive to create as it is to maintain, not least because of its tacit nature [33], easily lost at the point of need through staff movement and attrition.

Even before current industrial trends, many have recognised the importance of expertise expressed in products, processes and practices, but few have been able to actively bottle it: Design rationale [25], it has been suggested, bridges the information gap between the need a system fulfils and its final design. Often described as a snapshot of the decisions taken in reaching the final design [5], it tries to

capture the reasoning and knowledge that justify the resulting design [41], including arguments behind the choices made. As such, it may appear that design rationale is *the* tool for making explicit some valuable tacit knowledge during software development. However, design rationale capture is often not practiced as it can be time-consuming and expensive [5] and often places extra burdens on resources with no perceived immediate benefit [26]; specifically, 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. Therefore, it remains the case that much work on rationale capture and management remains theoretical. A greater uptake may be achieved by demonstrating to design practitioners that rationale management can be achieved as a by-product of design, rather than a bolt-on activity—particularly by providing efficient ways of classifying information and easy methods to create, navigate and retrieve knowledge—and that in the context of globalised IT Services processes can be improved through design rationale practices.

In this paper, we report on the conclusions of a study investigating the viability and benefits of capturing design rationale as a by-product of design in the context of a real-world global organisational setting. In the study, we tested the following ideas:

- That a principled approach can help practitioners identify the most relevant knowledge to capture as part of their design processes, contextualised with the needs of their own organisation. The principles we will exercise in this study are those underlying *Problem-Oriented Engineering* (see, for instance, [21,22], POE), a general framework for engineering, including design, proposed by the second and third authors. We will argue that some of those principles are equally beneficial as guidance to capturing designed artefacts (the outcome of design) and the decisions made during design and their justifications (essential elements of design rationale).
- That those principles should be embodied in practical techniques, which should not require practitioners to abandon their tried and tested processes and practices, but should instead strengthen and augment them whenever appropriate. A number of POE-based tools and techniques have been defined and applied to solving design problems in various domains. In this study, we will argue their suitability for design rationale capture alongside designing solutions. We will also argue that their combination with existing practices is facilitated by the very nature of POE, which is not a prescriptive method, but instead embodies a collection of conceptual tools for analytic (the understanding of a whole through its parts) and synthetic (the understanding of how each part contributes to a bigger whole) thinking.

The paper is organised as follows. Section 2 provides some background on design rationale management and on POE. Section 3 gives an overview of the study and its industrial context. Details of the study are given in Sects. 4–7. An evaluation of the study and its results is given in Sect. 8. Related work is discussed in Sect. 9, while Sect. 10 concludes the paper.

2 Background

2.1 Design rationale management

Design Rationale in its simplest form is [25]:

> "…the explicit listing of decisions made during a design process and the reasons why those decisions were made."

For software, it should also capture how a system 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 [17,28].

The importance of capturing design rationale in software engineering has been recognised for some time [25], particularly 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. With knowledge needed to support such systems distributed between many people, the likelihood of loss due to staff attrition is high. 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 [41]. Notable examples include standards on documenting software architectures, including the *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* [10]. It should be noted that these standards focus primarily on documenting particular aspects of a software solution, while in our study we take a broader view of design rationale management by also including design processes pertaining to the understanding of customers' needs prior to artefact design.

Capturing knowledge is a step crucial in avoiding loss [39], with its value to an organisation realised when available in a reusable form [5,7]. A typical example is the evolution of a software architecture due to market pressure or new customer requirements, or simply to develop a new system upon it. In such cases, the information embodied in the architectural design itself is often insufficient to permit direct reuse

[26], and it is a costly burden to record sufficient design rationale for reuse. Moreover, practitioners can become resistant if they feel the process of recording is too intrusive [5]. In fact, time and budget constraints on the capture of design rationale was found to be the most common barrier to documenting rationale in a recent survey of design practitioners [41]; the lack of tools to facilitate the capture of design rationale was also cited as a hindrance to recording the information necessary for the practice. In our study, we investigate the extent to which design rationale capture can be achieved as a by-product of design activities, rather than as an intrusive bolt-on activity. The approach we take is that of identifying specific points in the design process where such extra activities deliver immediate value to a project, as well as offering the promise of likely future reuse value.

The knowledge assets of an organisation include the tacit knowledge of experienced task experts [7], which must be elicited and recorded in a way that can be accessed and used by others. Important issues include choosing the representation that the design rationale should take [15], which has an impact on its reusability [25,28,39], how to recognise what constitutes a design decision, and how and when one should be captured [25,28,41]. In our study, we exercise some principles and techniques from an emerging design framework, to provide guidance as to which knowledge should be captured and at which point of a design process. Notably, we are not applying a one-size-fits-all approach: instead our techniques allow practitioners to analyse their own professional context, and to identify critical points in their processes in which knowledge recording would be of most benefit to their own organisation.

Douglas [15] suggests 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; and the decision itself. Regli [39] identifies two main approaches as having emerged to aid the capture of design rationale: process-oriented approaches, which capture the rationale as a history of the system being examined; and feature-oriented approaches, which focus on the representation of artefacts in the system. In [6], the authors argue that 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. To summarise, there is some consensus in the literature as to the key information elements involved in rationale capture. These are: decisions, i.e., the design choices made; justification for decision(s) taken, i.e., the reasoning, deliberations, criteria for selection and any related argumentation; alternatives, i.e., the various trade-offs; traceability, i.e., the relationships between layers of information; contacts, i.e., all relevant stakeholders. As we will see, such elements are present in our approach, with the caveat that we are not proposing a specific process all organisations should follow, rather we make principles and techniques available to organisations to contextualise their needs for design rationale capture within their own design processes.

2.2 Problem Oriented Engineering and Assurance-driven design

Problem-Oriented Engineering (see, for instance, [21,22], POE) is an emerging framework for engineering, the creative, iterative and often open-ended undertaking of designing and building products, systems and processes. POE sees engineering as a problem solving process in which interlocking *exploration* and *validation* steps are carried out: exploration of knowledge and its representation; and validation of represented knowledge. Both involve stakeholders: *finders* contribute to exploration; *validators* contribute to validation. Following an engineering tradition [40], POE problems concern the fundamental engineering question of how a solution can be designed to meet the requirements of stakeholders in a real world context. Problem solving is then a process of discovering relevant knowledge pertaining to those problem elements, and from that synthesising the solution.

Assurance-driven design ([20], ADD) adds a process view to POE, which recognises the need for ongoing management of the risk of inadequate validation during the development process to supplement (traditional) product-driven design methods. This risk that ADD explicitly addresses include: solving the wrong design problem and producing an inadequate solution (where inadequacy is defined with respect to all stakeholders).

Proposed by the second and third authors, POE and ADD have been successfully applied to the design of, and design process improvement for, software intensive systems, particularly in the area of high assurance and mission critical systems [18,30]. (Comprehensive and detailed introductions can be found in [20–22].) Without loss of generality, in the following we will use the term POE to indicate the application of the framework in combination with its ADD process view.

In the next section, we will discuss how elements of the approach were applied for design rationale management in our study. This work contributes to a long standing programme of research led by the second and third authors encompassing both the theoretical underpinning and the application and validation of POE in the context of software, information systems and other forms of engineering. As such,

it provides a first attempt at evaluating how POE performs for design rational management in a real-world setting.

3 Overview of the study

3.1 The method

In this study, we take a case study approach [43], in which rationale capture is investigated in the context of an organisational process which includes elements of design. The process is one which is used by the organisation in the resolution of customers issues arising from the use or deployment of information systems, and one which is typical across the sector in software supplying organisations. A case study approach was deemed appropriate due to the need to investigate design rational capture within the context of a design process and in the natural organisational setting in which the process is exercised. Therefore, a short-term study was conducted concomitant to the exercise of the process in the resolution of a specific customers issue, and within the normal practice of the organisation. The study was driven by one of the main stakeholders of that process (the first author), with the support of two external researchers (the second and third authors), who are POE experts. The specific customers issue investigated was typical of those routinely addressed by the organisation through their process. Data were generated both through: an initial questionnaire, distributed to stakeholders throughout the organisation and aimed at characterising the organisation context in terms of current design rationale capture practices and attitudes; a collection of documents, developed during the exercise of the issue resolution process; and a review with key stakeholders of the resolution process at the end of the study. A qualitative analysis of the data was performed by the authors, where an interpretative approach [34] was taken. An evaluation of the study and our findings is included in Sect. 8.

3.2 The organisational context

The context of the study is the first author's organisation, a UK-based subsidiary of an American financial institution, with business, systems and technical analysts based in the UK, technical architects in the US and development staff in India. The author's organisation—*the Software Supplier*—supplies a Mortgage Servicing Software package to the financial institution—*the Client*—a product that manages loan accounts once mortgage payments have been made by the Client's customers. The software facilitates business tasks such as payment calculation and processing, account queries, early redemption, correspondence, interest rate change and customer billing. The Software Supplier also provides support and assists in the resolution of issues that arise during the use of the supplied software.

Recently, the Software Supplier has lost a number of subject matter experts but has retained a contractual obligation to provide support to the Client to enhance and maintain the supplied software stack. This has motivated the Software Supplier to investigate through this study the capture of design rationale as a way to mitigate the risk of loss of key knowledge assets.

4 Initial questionnaire

An initial questionnaire was designed to help us characterise the organisational context of the study and to gather information related to practices and perceptions of staff in the Software Supplier with regard to design rationale. The questionnaire design was influenced by findings in the literature, particularly [28,39,41], and also informed by detailed knowledge of the first author related to recent organisation's off-shoring and restructuring processes. A questionnaire was chosen over face-to-face interviews as it was considered the most cost-effective and efficient means of collecting data from staff members globally dispersed over three continents and within the time constraints of the project. Closed questions using Likert-style scales were used, combined with the option for respondents to provide discursive comments.

Although the number of respondents was small (9 out of 15 invited), their spread within the organisation in terms of diverse functions and years of IT experience was considered sufficient to provide a suitable representative sample within the division of the organisation in which the first author works, which is made of 10 onshore (UK), 20 offshore (India) and 3 offshore (US) staff. All of the respondents had at least 2 years experience in IT and two of them had over 10 years. The respondents covered the following job functions: Development, Support, Business and Systems Analysis, Product assurance (Testing), and Project and Senior Management. Here, we summarise the results briefly; a detailed analysis of the responses can be found in [32].

All respondents stated that they agreed or strongly agreed with the statement that having access to design rationale would help them in their work and benefit the organisation.

Respondents were asked how often certain sources of information were referred to, when they needed some background information. The responses indicated that original project documentation (within the organisation this would primarily be requirement documents) and subject matter experts were most often referred to. Other sources listed on the responses included: history (of the application); subject matter documents; and emails between relevant parties.

Respondents were also asked what they felt affected their ability to reuse design rationale information. The results

indicate they 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 (business analyst) 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.

Among the factors that were perceived to affect the capture of design rationale in the organisation, time constraints appeared to be a major factor (this is in line with findings in [41]). The responses also showed that not knowing what information should be documented affected the recording of rationale (this was also noted in [39]). Other factors suggested included: type of solution, i.e., 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, leading to things being missed or their significance misunderstood; methodology and QA Process.

The outcome of the initial survey helped us identify a suitable organisational process for in-depth study. Specifically, a key process of the Software Supplier, the Client's issue resolution process, was chosen and exercised on a particular Client's issue, that of correcting billing errors in mortgage calculations. The chosen Client's issue resolution process had the attractive characteristics of involving stakeholders across the traditional problem/solution divide, that is from clients and business analysts all the way through to software architects and developers, hence addressing the wide spectrum of activities where design rationale capture had been identified as critical by the initial survey. The chosen process is also a typical process in software supplier organisations, being one of the key processes identified in widely adopted IT Governance frameworks (e.g., the guidelines on incident resolution in ITIL [24]), hence a promising springboard for future possible generalisation of the study to other organisations. Appropriate POE principles and techniques were applied by the first author, in consultation with the co-authors as POE experts, in the course of a live project in which the billing errors issue was considered and resolved. As a member of the *Supplier production support team* (SPST), the first author could secure adequate access to background information, documentation and the stakeholders for the issue. The problem also needed a resolution that involved some design activity and therefore provided an opportunity for design rationale management. Both the generic process and its particular application to the billing errors issue were considered as part of the study. Specifically: in the study of the generic process (cf. Sect. 5) aspects of design rationale management including key process activities, artefacts produced in the course of the process and the stakeholders involved, were considered; in the study of the specific resolution of the Client's issue (cf. Sect. 6), further aspects of design rationale management were studied, including specific design steps and decisions, their justifications, validation criteria and trade-offs, and overall traceability.

5 The Client's issue resolution process

In this section, we will present in detail the application of POE to the generic process under study, leading to a POE characterisation of the process itself.

5.1 The process

The Client's issue resolution process used by the Software Supplier at the time the study was conducted is described below. Relevant key terms and stakeholders are summarised in Tables 1 and 2.

When an issue is found in the Client's use of the Software Supplier's applications, a *Triage Document* is raised by the *Client production support team* (CPST) to describe the problem with information included that may assist in tracking down its cause. The reported issue is given a priority by the Client (*low*, *medium*, *high*) that governs the timeline for response and solutions, based on service-level agreements. Once the *Triage document* is received by the *Supplier production support team* (SPST), 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.

Further discussions may be held between the Client and Supplier production support teams to agree: (a) which issues lie with the application software; and (b) an approach for dealing with the issue. Additional clarification may be sought from the Client from which the issue report originated. The

Table 1 Summary of key terms

Term	Description
Mortgage Servicing Software	Software used in mortgage servicing activities, such as interest accruals, billing, collecting due payments, redemption of loans etc
Triage document	A form used to report details of a production issue
Financial Services Authority (FSA)	Regulatory body for financial institutions

Table 2 Summary of key stakeholders

Stake-holder	Description
Client	The financial institution managing customer mortgages
Customers	Patrons of the Client whose mortgages are being managed
Software Supplier	The author's organisation that supplies and maintains mortgage servicing software for the Client. The case study is based in this organisation
Client Production Support Team (CPST)	Group of individuals on the Client side tasked with dealing with issues raised in regards to application software being used in day to day business activities. They keep contact with the Client, understand the workings of the application system and the platform that it runs on, provide the initial information on issues and communicate with the Mortgage Business Management within the Client when questions arise of a business policy nature. They also communicate with Senior IT Management on the Client side in case of issue escalation
Supplier Production Support Team (SPST)	Groups of individuals on the Software Supplier side tasked with dealing with issues raised in regards to application software being used in day to day business activities. They keep contact with CPST, assign work to the development and mange releases of solutions to the Client. They also communicate with Senior IT Management on the Software Supplier side in case of issue escalation
Application Architect (AA)	Reviews a solution to assess if solution complies with standards
Offshore Development Team (DT)	Group of individuals tasked with developing software
Product Assurance Team (PA)	Ensures the quality of the provided solution

clarification may be in the form of screen shots of the application error, data extracts, event logs and example scenarios.

When issues are agreed between CPST and SPST, they are analysed and solution approaches proposed by the development and architecture resources assigned to the issue. The proposed solutions are discussed with the CPST. Once agreement is reached on a solution approach, it is developed and tested. On completion of development and testing, the solution is packaged by the SPST with release notes and a test report, and delivered to the CPST.

Subsequently, the CPST validate the delivered package, perform some further tests in collaboration with the Client, and may either return it for rework if it is unsatisfactory or implement it to the production systems if satisfied with the results.

5.2 POE process principles and techniques

Design in POE is based on the notions of "problem" and "problem solving." The POE process view is summarised in a pattern (see [20] for a detailed treatment) which distinguishes and juxtaposes the following:

- *Activities* In *problem exploration* knowledge of the context and requirement of the problem (or part thereof) is captured. In *solution exploration*, knowledge of the solution (or part thereof) is captured. These activities are partial as problem solving may (initially) focus on parts of the context, requirement or solution, rather than on the whole problem.
- *Stakeholder roles* The roles are of *Problem Validator*, *Solution Validator*, *Problem Finder* and *Solution Finder*. A Problem Validator validates a (partial) candidate problem descriptions. Examples include a client, a customer, a regulator. A Solution Validator validates a (partial) candidate solution descriptions, and may be, for instance, a chief engineer or a project manager.
- *Choice-points* The choice points allow validation to affect exploration in determining whether from a current candidate characterisation of the problem the solution may be investigated, and whether from the current candidate solution further problem exploration is appropriate or that backtracking the process—to find another candidate solution or to explore the problem further—should be done.

The design principles embodied in the POE pattern are:

- activities that pertain to exploring the problem should be separated out from those pertaining to exploring the solution;
- check-points should be associated with exploration activities, to handle risk. The primary risks are of addressing the wrong problem or of designing the wrong solution for a validated problem;
- 'finder' and 'validator' are two separate roles, with finders involved in explorations and validators checking the outcome of explorations. Finders and validators come together at check-points, allowing risks to be transferred from finders to validators.

These principles can be applied in analysing and modelling existing processes, possibly leading to process improvement as we discuss in this section. Processes which are amenable to this type of analysis are problem solving processes, typically underpinning design and engineering. These principles can also be used to drive the process of design of specific artefacts (as we will discuss in Sect. 6).

In the analysis and modelling of an existing process, the application of the pattern and its related principles requires that process activities, roles and check-points be suitably identified, classified and captured in the model. The separation of problem and solution activities helps one to uncover any possible bias towards either problem or solution, while an evaluation of the number of frequency of check-points expose resource-risk trade-offs embedded in the process. Such an analysis can help one identify required interventions for process improvement.

5.3 Application of POE to the Client's resolution process

During the study, a POE analysis was applied to the Client's resolution process leading to the model of Fig. 1. In the model, based on Business Process Model and Notation (BPMN) [35]:

- rounded rectangles represent activities in the process. As a result of its POE analysis, we have colour-coded the activities so that those with a while background correspond to problem explorations and those with a grey background to solution explorations. One activity, "Prepare problem specification document," has some grey shading to indicate that it is a combination of problem and solution exploration (we will return to this activity in the discussion below).
- swim lanes represent stakeholders involved in the process. Activities within a swim lane are interpreted as carried out by the corresponding stakeholder. (We have taken the liberty of stretching an activity across swim lanes to indicate that the activity is carried out in co-operation by the corresponding stakeholders. This is a departure from standard BPMN, which does not appear to model the sharing of activities among stakeholders.) From a POE perspective, stakeholders conducting problem exploration activities assume corresponding 'finder' roles.
- diamonds represent check-points. Through POE analysis we have located each check-point in the swim lane of the stakeholder who is the 'validator' for that check-point.
- concrete artefacts, like documents or software, are indicated as dog-eared icons, with the dotted arrows indicating the activities which produce and make use of the artefact. Labels indicate the nature of the artefact and its status at a specific point in the process. Notable among them is the "Problem Specification Document" which was not part of the original organisational process, but was added as a result of the study (see discussion below).

As part of the modelling, process activities were analysed and separated into their exploration and validation components, and finder and validator roles were attributed to relevant stakeholders. It was observed that in the original process Client's validation was missing between the initial validation of the Triage document and the validation of the implemented solution. This was found to be a gap in the Software Supplier's 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 proposed solution also allowed the Client to introduce changes at late stages of development and reject solutions as unsuitable after delivery.

As a result, a new document was introduced during the study, the Problem Specification document, to fill this gap. The document, designed to include the validated problem description, analysis and clarifications, estimates and the proposed solution, created visibility of the proposed solution and related information to relevant stakeholders before significant development work was carried out. The related activity to create such an artefact was classed as a mix of problem and solution exploration because it required both knowledge of the problem and the proposed solution to be captured.

6 The problem of billing errors in mortgage calculations

In this section, we will present in detail the application of POE to the resolution of a specific customer issues, and the documentation generated as a result of combining POE techniques in the context of the normal practice of the organisation.

6.1 The problem

The problem chosen for our study, which we call the *Valid Amounts* problem, concerns the correction of billing errors in mortgage calculations. Briefly, customers are billed 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 Financial Services Authority (FSA), the Client must be able to explain the reasoning behind their charges and services and also provide their customers with information on these factors sufficient to remain compliant. The impact of such errors, that the Client can be held in breach of statutory regulation, makes their

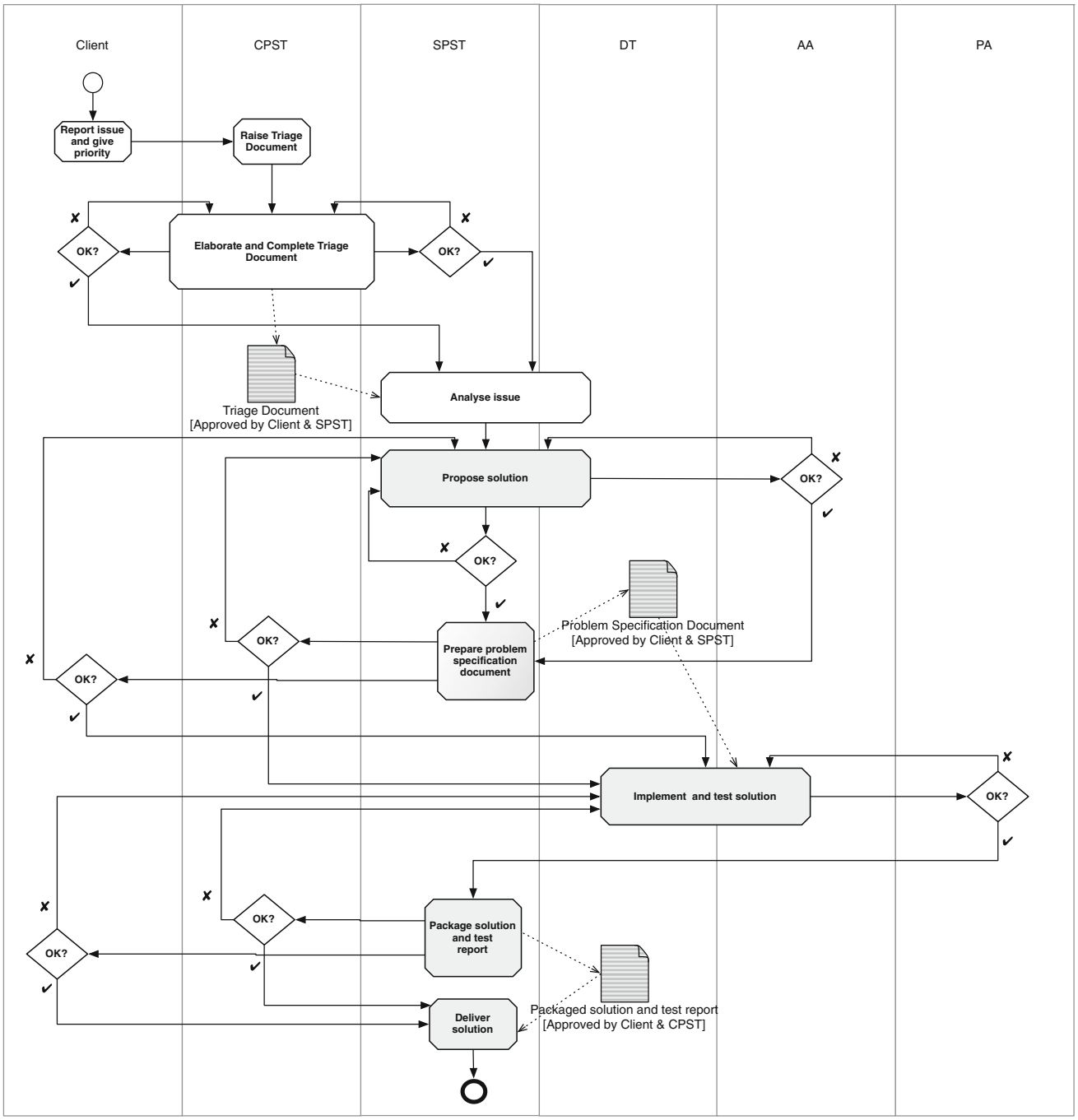

Fig. 1 The Client's issue resolution process modelled on the basis of the POE process pattern and related principles. Notation: swim lanes represent stakeholders, rounded rectangles represent activities, diamonds check-points, and dog-eared icons artefacts

resolution high priority. The resolution of such issues is assisted by the Software Supplier, who may deliver code and/or data fixes.

6.2 POE design principles and techniques

As discussed in Sect. 5.2, as well as analysing existing problem solving processes, the POE process pattern and its principles apply to drive artefact design, which is also seen as a form of problem solving.

The notion of 'problem' at the heart of POE follows an engineering tradition [40] being concerned with the fundamental engineering question of how an artefact (the solution) can be designed to meet the needs (the requirement) of stakeholders in a real world domain (the context). Design as problem solving is then a process of capturing relevant knowledge pertaining to requirement and context, and from

that synthesising the solution. This process is step-wise: each step being an exploration which contributes in some measure to knowledge capture and/or solution synthesis. Each step is accompanied by validation activities: the generation of evidence which can be used by stakeholders at check-points in the process. Successful validation advances the process by increasing the level of assurance that an adequate solution may be reachable; unsuccessful validation backtracks the process requiring the elements of the problem or any step taken to be rethought.

Together with the already stated POE process principles, the following principles are embodied in POE problem modelling and solving:

- a problem model is made of three fundamental descriptions pertaining to the elements of a problem: the descriptions of a real-world context, a requirement and a solution.
- problem descriptions are captured as part of the process in a step-wise fashion,
- the nature and form of the descriptions should be parameterised by the specific design needs and context in which the process takes place,
- each design step should be justified: its justification should collect evidence that can be used by validators at check-points in the design process,
- the form of the evidence collected at each design step should be parameterised by the specific validation needs and context in which the design process takes place,
- validation activities tend to lower downside risks and increase resource expenditures.

As mentioned in the principles above, POE does not impose any particular form of description or justification, but does provide a set of techniques to help one structure and track them. Those introduced by [18,19,38] were used in the study, as reported below, to represent problems, their transformation, justifications and related validation activities.[1] Briefly, they are:

Problem diagrams

Problem diagrams provide a graphical notation for sketching problems, which is based on Jackson's problem diagrams [23]. Their elements can be used to represent: the requirement, the solution and the collection of domains constituting the relevant parts of its real-world context, together with ways they relate to each other through phenomena of interest. Those elements are illustrated in Fig. 2; each requires an associated description. The representation of a solution

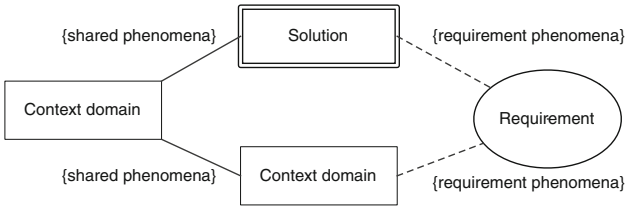

Fig. 2 POE problem diagram notation: context domains as undecorated rectangles (two in the example); solutions as decorated rectangles (there is only one in each diagram); requirements as ellipses (there is only one in each diagram); phenomena shared among domains, or domains and solutions, and phenomena mentioned by requirements as arc annotations

artefact can also include an indication of the elements that make up that solution as illustrated in Fig. 6.

Trasformation diagrams

Transformation diagrams provide a graphical notation for sketching design steps in the form of problem transformations, that is the way a problem is transformed into others as a result of a step. An example is given in Fig. 7, which illustrates that the problem diagram below the horizontal line is transformed into that above the line as part of the design step; the corresponding justification (J_2) is indicated to the right of the line, while to the left is a representation of the validation that has taken place for that step. By convention, in a transformation diagram problem validation is represented to the right, while solution validation to the left. POE defines a set of standard transformation classes through which problems can be transformed, as detailed in [21,22]; these were applied in the study as part of the design process.

Design trees

Design trees provide a graphical notation for representing the steps taken as part of the overall design process. One such tree is illustrated in Fig. 3 for the problem under study. Essentially a design tree is a compact representation of the way the various transformation diagrams are related in the design process, together with an indication of where the process started, ended and any backtracking that may have occurred.

Justification templates

A template for capturing a justification associated with a design step was introduced in [18], as a result of some empirical work on applying POE in the design of safety critical systems. The same template was used in this study as the means to record, among other information, design rationale, identified risks and validation activities involving relevant

[1] The formal mathematical notation of [21,22] was deemed inappropriate in the business setting of the case study.

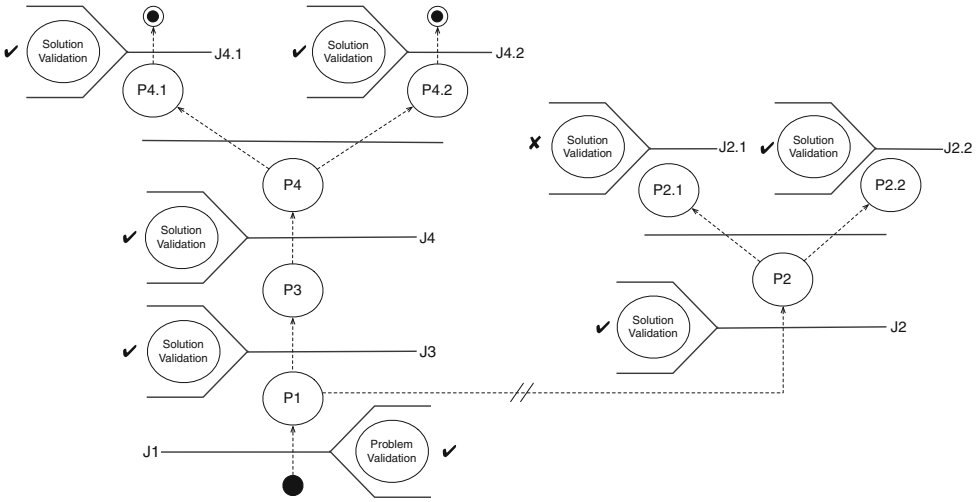

Fig. 3 Design tree for the Valid Amounts problem: to the *left*, the successful design tree; to the *right*, a prior unsuccessful branch which led to backtracking

Table 3 Problems

Name	Description
P1	The initial Valid Amounts problem
P2	The Valid Amounts problem with two solution sub-components (Mortgage Processing (Calculations) (P2.1) & Loan Balances (P2.2))
P2.1	The Mortgage Processing (Calculations) co-design sub-problem
P2.2	The Loan Balances co-design sub-problem
P3	The backtracked Valid Amounts problem with one solution sub-component (Loan Balances (P3))
P4	The Loan Balances Problem with two solution sub-components (Expected Values (P4.1) & Adjustments (P4.2))
P4.1	The Expected Values co-design sub-problem
P4.2	The Adjustments co-design sub-problem

stakeholders. Many examples of the template application are provided below.

6.3 Application of POE to the Valid Amounts problem

POE was applied to the Valid Amounts problem in the context of the Software Supplier's established Client's issue resolution process. In Fig. 3, a POE design tree provides a compact overall view of the steps followed in the study to reach a successful design: on the left hand side there is the successful design tree; on the right there is a backtracked branch which was followed first but that did not result in an adequate design. The various problems Pi addressed in the study (e.g., P1) are summarised in Table 3, with derivation of one from another indicated by dotted arcs, starting at the bottom, where the black dot indicates the start of the process, and proceeding towards the top. Horizontal lines indicate POE transformations, annotated with their justification (e.g., J1) and related validation activities; a tick next to a validation activity indicates that it was successful: there is a single unsuccessful validation requiring backtracking in the right branch; this is described more fully later in this section.

In what follows we will provide a detailed presentation of the initial transformations in this process in order to illustrate how POE was used for design rationale management. We will then provide a brief summary of the remaining steps; for the interested reader, a more detailed description of the study can be found in [32].

6.3.1 Problem exploration

The purpose of problem exploration is to capture details of the problem, its context and requirements. The initial problem details were provided by the CPST in the form of a Triage Document describing the problem and other useful information. As a member of the SPST, the first author

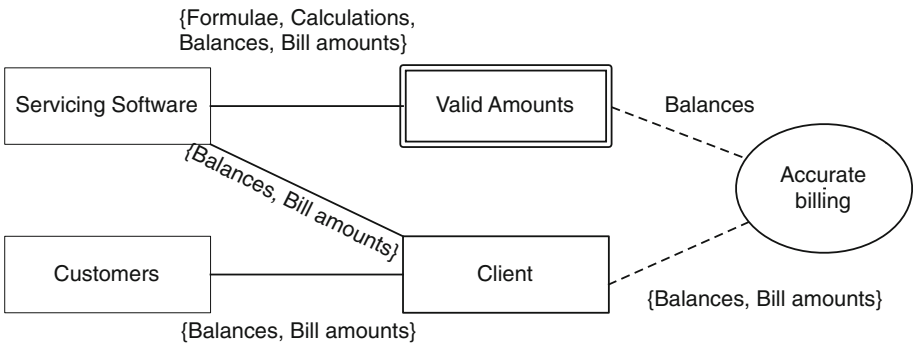

Fig. 4 P1: The initial Valid Amounts problem model

checked details in the Triage Document to establish that it was adequate as the basis of further investigation. Additional examples of the issue and clarifications were obtained from the Client. Problem exploration was used to examine the information on the Triage Document by the SPST. By applying POE domain and requirement interpretation transformations [21,22] the problem model shown in Fig. 4 was derived.

POE principles recommend that justification obligations be discharged for each POE step: these correspond to various design concerns which arise in the execution of a step. Examples include the validity of descriptions introduced with respect to their real-world counterpart, and the feasibility of a chosen solution architecture. Justifications record such concerns together with all identified risks, some evidence of the action(s) taken to address them and any related stakeholder validation sought.

The justification for the interpretations leading to P1 is given below, obtained by applying the template of [18] to the information in the Triage Document: descriptions and phenomena of the problem, and concerns, claims and evidence of their validity are included in the justification. The main concerns emerging during problem explorations have to do with the validity of the context and requirements descriptions. Context validity is about making sure that our understanding of the context correspond to reality, so to avoid erroneous assumptions on which the designed solution may rely upon. Therefore addressing this concern means managing the risk of neglecting or misunderstanding context properties, or making unwarranted assumptions. Requirement validity is about making sure that our understanding of the requirements corresponds to the real need, that is managing the risk of addressing the wrong problem.

STEP ID: *Initial* PROBLEM EXPLORATION *leading to P1*

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: Here are the initial context and requirement descriptions:

Name	Description
Accurate billing requirement	That customers are correctly billed for the mortgage loans that they have with the Client
Valid Amounts	The solution that is needed to ensure that customers are accurately billed
Client	The financial institution managing customer mortgages
Customers	Customers whose mortgages are being serviced
Servicing Software	Software used for servicing mortgage loans

and here are their phenomena:

Name	Description
Calculations	Application of interest and other factors to produce a figure that the customer should be billed for their loan
Bill Amounts	The amounts the customer is advised to pay for a named period (usually monthly)
Balances	Total amounts for each mortgage loan
Formulae	Formulae used to calculate due amounts

CONCERN: Problem validity
STATUS: Pending

CLAIM: This is a valid initial characterisation of the Valid Amounts problem, its context and requirements.
RISKS: Insufficient or inaccurate information provided for problem solving.

The validity concern—indicated by "CONCERN" in the template—raised during exploration, remained in the *pending* state until problem validation was concluded, discussed in the next section.

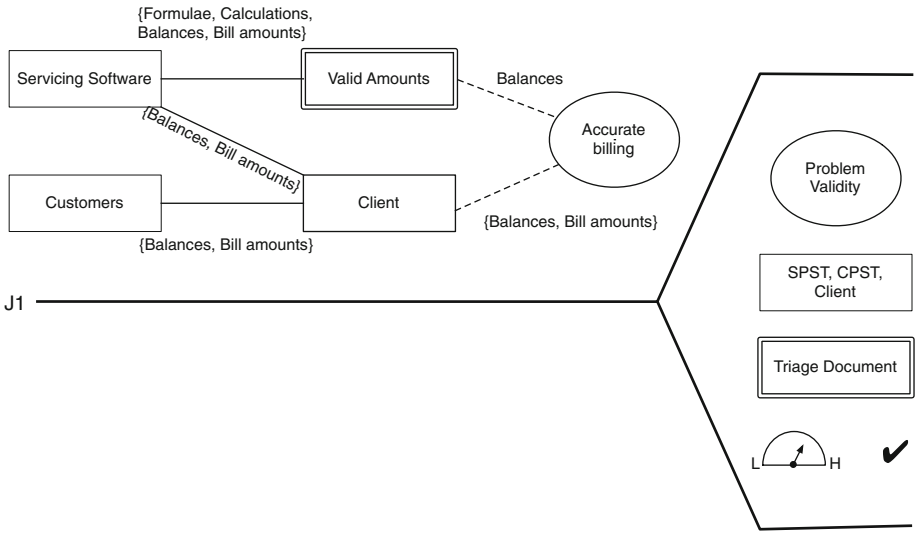

Fig. 5 Problem validation of P1, with stakeholders SPST, CPST and the Client, and the Triage Document as key validation artefact

6.3.2 Problem validation

Problem validation transfers the risk of misunderstanding a problem from Problem Finder to Problem Validator; thus is an agreed problem-to-solve arrived at. The consequences of an unvalidated, or an incorrectly validated, problem description could be a solution that solves the wrong problem. This might impact Client confidence, for instance that the Software Supplier can provide adequate solutions, and have resources implications, *viz*. the time and effort expended designing, developing, testing and delivering the wrong solution. 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; problem validation acknowledges this shared understanding.

Problem validation 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. The SPST reached the initial description of the Valid Amounts problem from the Triage Document, clarification emails and face to face meetings with the Client and CPST. The Triage Document was reviewed by the SPST, and where necessary, gaps in the provided information were dealt with through follow up.

STEP ID: PROBLEM VALIDATION *of P1*

CONCERN: Problem Validity
STATUS: Discharged

CLAIM: This is a valid initial characterisation of the Valid Amounts problem, its context and requirements.
ARGUMENT & EVIDENCE: The information provided on the Triage Document was examined by the SPST and additional clarifications supplied by CPST and Client as requested. The information was assessed and deemed sufficient for further investigation. The context and requirement interpretations were deemed to represent adequately the information as reported in the Triage Document.

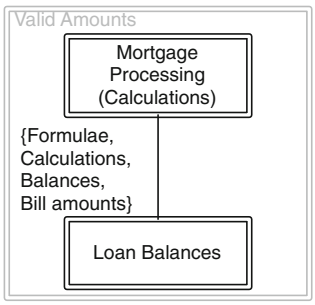

Fig. 6 Solution interpretation of Valid Amount into two components

RISKS: Insufficient or inaccurate information provided for problem solving.

Figure 5 shows the validated initial Valid Amounts problem. The stakeholders involved were SPST, CPST and the Client, with the key validation artefact an adequately populated Triage Document. The symbols represent: the level of risk, on a scale low to high; and the status of the validation, discharged in this case. This convention is used to represent validation steps in the case study.

The risk in this step is shown as fairly high as there is a high risk of the problem not being correctly understood if the Triage Document is unclear or inadequately detailed. This risk must be dealt with before progressing to issue resolution.

6.3.3 Solution exploration

Consideration of the solution led the Valid Amounts domain to be structured into two related components, Mortgage Processing (Calculations) and Loan Balances, as shown in Fig. 6 and captured in the step justification below. The main design concern in this step is solution validity: making sure that the chosen solution can meet the established

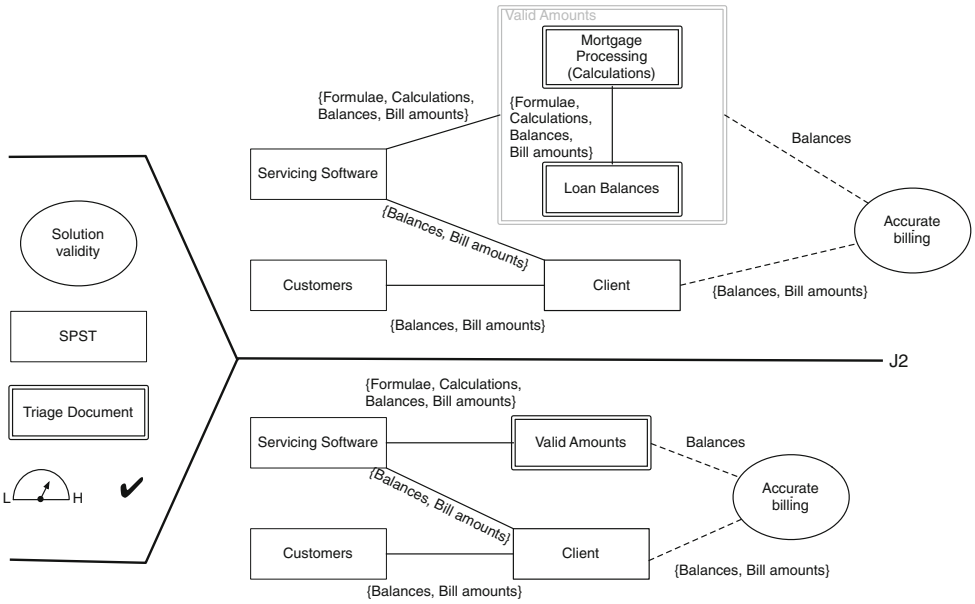

Fig. 7 Solution exploration and validation leading to P2

requirements in context, hence mitigating the risk of designing an artefact that will not solve the problem.

STEP ID: SOLUTION EXPLORATION *applied to P1 leading to P2*

JUSTIFICATION J_2: 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. There are two parts to the valid amounts problem:

- looking at the calculations in mortgage processing: if any formula is not correct for particular scenarios (business rules, conditions) then the result will be invalid amounts for billing; and
- looking at the loan balances: 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.

DESCRIPTIONS & PHENOMENA: Following from the above, the corresponding solution components and their descriptions are:

Name	Description
Valid Amounts	The solution that is needed to ensure that customers are accurately billed, made of two components: Mortgage Processing (Calculations) and Loan Balances
Mortgage Processing (Calculations)	The processing, rules and formulae applied to the loan balances on a loan account
Loan Balances	Amounts held for a loan account that are used to carry out calculations: principle balance, interest rate, arrears amounts, prepaid amounts, etc

and here are their phenomena:

Name	Description
Calculations	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
Bill Amounts	The amounts the customer is advised to pay
Balances	Financial figures relating to loan accounts
Formulae	The methods and calculations applied to generate billing amounts and update account balances

CONCERN: Solution Validity
STATUS: Pending

CLAIM: This is a valid solution to satisfy the requirement in its context, that is an implementation based on this design will lead to valid amounts being billed to customers. This is necessary for the Client organisation to ensure that Customers remain satisfied, and compliance requirements of the industry ombudsman, in this case the FSA, are met.
RISKS: If the requirement is not met: Customers will continue to be charged incorrectly and are likely to become dissatisfied; and Customers will not be charged in line with the terms that they have been told, leading to compliance issues for the Client.

6.3.4 Solution validation

Architectural validation discharges the concerns related to this architectural choice by exposing evidence and arguments to relevant stakeholders. Architectural validation was obtained addressing the validity concern as indicated in the step from P1 (initial problem) to P2 (chosen solution architecture) in Fig. 7 and detailed below. In this case, validation was performed the SPST using the Triage Document as the main artefact to guide validation of the architectural design.

STEP ID: SOLUTION VALIDATION *of P2*

CONCERN: Solution Validity
STATUS: Discharged

CLAIM: This is a valid solution to satisfy the requirement in its context, that is an implementation based on this design will lead to valid amounts being billed to customers. This is necessary for the Client organisation to ensure that Customers remain satisfied, and compliance requirements of the industry ombudsman, in this case the FSA, are met.
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. Separating the problem in this way allows for the root cause (entailing application code fix) and effect (entailing data fix) of the issue to be dealt with individually if required, and is in line with normal issue resolution practices. Both aspects need considering in order to solve the Valid Amounts problem, as fixing the calculations may not result in the balances being corrected and fixing the balances may not prevent reoccurrence of the issue. Therefore, the two parts of the solution require some co-design.
RISKS: If the requirement is not met: Customers will continue to be charged incorrectly and are likely to become dissatisfied; and Customers will not be charged in line with the terms that they have been told leading to compliance issues for the Client.

6.3.5 Further solution exploration

Further solution exploration of the identified two solution sub-components was carried out, leading to two corresponding sub-problems: P2.1 and P2.2 of Fig. 3. We briefly discuss such sub-problems in this section and explain how their validation process led to backtracking of the design process. Only extracts of the steps are presented here, while full detail of the analysis can be found in [32].

Exploring the Mortgage Processing sub-problem (P2.1), resulted in further decomposition of the solution domain into two components, corresponding to the two processing modes which can be used to change loan accounts, that is:

- Interactive: carried out by a user via screen entry; and
- Batch: bulk handling of accounts offline.

In the provided triage information, some functional areas were identified for further investigation as possible root causes of the problem. These were: manual adjustments and redemption activities in interactive processing; and billing in the batch processing.

Here is a summary of the concerns which emerged during solution exploration and which address both the primary risk of an invalid solution, and the secondary risks of compliance and customer satisfaction.

STEP ID: SOLUTION EXPLORATION *of P2.1*

[...]

CONCERN: Solution Validity and Feasibility
STATUS: Pending

CLAIM: A solution of the Mortgage Processing sub-problem would eliminate reoccurrences of the issue, hence, it is considered to be a "strategic" approach to solving the issue.
ARGUMENT & EVIDENCE: Addressing root causes and providing fixes to resolve them would stop the problem re-occurring. Root causes can occur either during interactive or online processing 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. Also, identifying commonalities between affected accounts could lead to automated resolutions for the problem. The advantages of this approach are: little or no manual intervention is 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 further intervention may be required as it may not actually fix the balances that are incorrect.
RISKS: Balances not fixed by correcting root causes.

CONCERN: Customer satisfaction
STATUS: Pending

CLAIM: This solution will result in reduced impact to customers.
ARGUMENT & EVIDENCE: Impact is reduced as 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 root cause is investigated and the solution being developed; amounts may still need correcting after the root cause has been addressed.

Exploring the Loan Balances sub-problem (P2.2) also led to two parts of the solution being identified:

- Expected Values domain, which encompasses the correct balance amounts provided by the Client's Spreadsheet; and
- Adjustments domain: the functionality to apply to 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.

STEP ID: SOLUTION EXPLORATION *of P2.2*

[...]
CONCERN: Solution Validity and Feasibility
STATUS: Pending

CLAIM: Solving the Loan Balances sub-problem would address the balance data held on accounts using a fix program to minimise the impact to customers in the short term, it is hence considered to be a "tactical" approach to solving the issue.
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. The advantages of this approach include: it 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, leading to a long projected timeline to fix all the identified accounts; possibility of introducing new errors; root causes are not addressed so customers may be impacted again, leading to the number of affected accounts increasing over time.

CONCERN: Compliance
STATUS: Pending

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: Root causes are not addressed so issue may re-occur.

CONCERN: Customer satisfaction
STATUS: Pending

CLAIM: This solution will remove impact to customers.
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.

6.3.6 Further solution validation

All concerns raised in the exploration of P2.1 and P2.1 were then addressed through solution validation.

The Solution Validity and Feasibility concern was discharged by the application architect (AA), who validated the two proposed solutions, and established that they addressed the problem (validity) and were within the capabilities of the company to deliver (feasibility); the AA also established that they conformed to standards. With reference to Fig. 3, this is the step with justification J2 and successful validation.

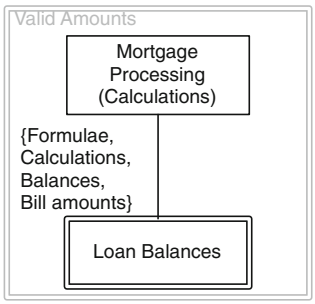

Fig. 8 Single component solution architecture for Valid Amount after backtracking

The two solution options were then presented for further validation to the Client and CPST. Note that this latter step was added to the process as a result of applying POE as discussed in Sect. 5.

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 compliance and customer satisfaction concerns and their potential risks. In particular, the Client concluded that: 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 a difficulty in determining if all root causes have been discovered and fixed; the time scales for discovery could prove to be unacceptably long. It was also observed that some of the root causes were being addressed under other reported issues, outside the scope of this problem. With reference to Fig. 3, this is the step with justification J2.1 and failed validation.

On the other hand, the "tactical" Loan Balances solution was accepted by the Client. In particular, this solution was deemed to resolve the compliance issue by: ensuring that balances are correct, with the Calculation Spreadsheet providing a documented audit together with a report of the changes made; and addressing customer impact 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 was considered acceptable. With reference to Fig. 3, this is the step with justification J2.2 and successful validation.

6.3.7 Remainder of the development

As a result of the Mortgage Processing (Calculations) strategic solution being discarded, the design process was backtracked to the initial Valid Amounts problem for which a single component solution architecture was then chosen, as illustrated in Fig. 8: compared to the previous solution

architecture of Fig. 6, this new solution assumes mortgage processing as a given, with only the Loan Balances problem to be addressed. As a side effect, the risks associated with the validity of a solution based on this new architecture are only partially mitigated: the residual risks of more customers being impacted since the root causes of the issue are not addressed remain, with an understanding that the Client is willing to accept such risks.

From this point onwards, the Loan Balances problem analysis was replayed in the same way as explained in the previous section, with its identified subproblems (Expected Values, P4.1, and Adjustments, P4.2) then successfully solved. This led to the conclusion of the design process, as indicated in Fig. 3.

7 End-of-study review

A review of the study was performed by the first author via an informal meeting involving other process stakeholders and her line manager, the Production Support Manager. During the review, both the artefacts generated in the study and their use in the resolution process were considered, with particular attention on how well POE practices combine with the normal practices of the organisation, their perceived intrusiveness or otherwise, and their perceived immediate and potential long term benefits. Specifically, it was noted that their application did not require a radical change in current practices; instead these could be exercised as usual, but augmented with POE related activities. For instance, the currently adopted Triage Document could still used as a key document for capturing the initial knowledge about a Client's issue; however, the application of POE principles helped with the identification of specific information related to the problem context, the requirement and the proposed solution to be included and validated in the form of the newly introduced Problem Specification document. In particular, it was remarked that the discipline of analysing the information contained in the Triage Document by mapping it to the POE problem model, helped one both with collecting transient information in a structured and traceable form, as well as with highlighting relevant concerns and arguments to be used in validation of those descriptions with stakeholders. Also, the ability of including information about the proposed solution design within the same model was considered advantageous as the same document could be used to relate information to different categories of stakeholders, from business to technical experts. The consensus was that such practices had added value to the process, to a point that a decision was taken to modify it in order to include such an analysis and corresponding validation with the Client and their support team as a matter of routine. In other words, the extra effort required to produce the new document was seen as a worthwhile and cost-effective investment which delivered value not just in that particular instance, but could potentially contribute to future process improvement and cost-savings. Although not directly experienced in the study, it was also felt that the information explicitly recorded in the document can contribute to design rationale reuse in that it is available to assist decision making should additional work be requested in the area, e.g., should the client decide to revisit the strategic solution in future occurrences of the billing errors.

8 Evaluation

In this section, we reflect on the outcome of the study in the light of the overall aim of the work—that of investigating the viability and benefits of using POE to capture design rationale as a by-product of design in the context of a real-world global organisational setting—and its specific hypotheses (recalled from Sect. 1):

- That a principled approach can help practitioners identify the most relevant knowledge to capture as part of their design processes, contextualised with the needs of their own organisation.
- That those principles should be embodied in practical techniques, which should not require practitioners to abandon their tried and tested processes and practices, but should instead strengthen and augment them whenever appropriate.

8.1 Helpfulness in capturing design rationale elements in an organisational context

Basic POE principles and techniques for process and problem modelling, and problem solving were applied throughout the study to help the first author and her team identify the most relevant knowledge to capture as part of design activities related to both the generic Client's issue resolution process and its particular application to the Valid Amounts problem. It was found that they were instrumental in guiding knowledge capture of key design rationale elements including knowledge of the designed artefacts (the chosen solution to the Valid Amounts problem), the design process, the decisions made during design and their justifications. Specifically, and with reference to standard design rationale elements from the literature (a summary is given in Table 4):

- *Contacts* The four roles in the POE process pattern ensure that relevant stakeholders are identified and involved in the process at appropriate steps. This is captured both in the process model and in the validation activities associated with problem transformations. Although for the purposes of confidentiality, specific names have not been

Table 4 Elements of Design Rationale and their mapping onto POE

Element	POE equivalent
Decisions	Design choices and their validation, explicitly captured through exploration and validation steps
Rationale for decisions	Claims, arguments and evidence associated with concerns in justifications
Alternatives and trade-offs	Solution exploration and backtracking
Suitability	Stake-holder validation of concerns identified in justifications
Constraints	Descriptions of problem elements; claims, risks, arguments and evidence associated with concerns in justifications
Assumptions	Descriptions of problem elements and validation artefacts
Status	Tracked in justification concerns
Complexity of the design	Problem diagrams, design trees, process pattern
Issue descriptions	Claims, risks, arguments and evidence associated with concerns in justifications; validation artefacts
Traceability	Problem diagram, design trees, process pattern
Contacts	Stakeholders roles, i.e., problem finder, problem validator, solution finder, solution validator

included for the stakeholders in the study, they have been identified by their role in the organisation involved. In a real life situation, the names and contact details would be noted alongside their role assignments.

- *Decisions* Design choices are recorded in POE validation steps as well as the descriptions of designed artefacts associated with problem models. An example of this is the selection of the tactical Loan Balances solution over the more strategic Mortgage Processing option in the solution validation step associated with P2.1.
- *Rationale for decisions* The POE justification template adopted in the study 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.
- *Constraints* Constraints are captured in design concerns, like the compliance concern, which imposed a constraint that had to be satisfied by the chosen solution. Other constraints may derive from properties of the problem context, explicitly captured in context descriptions.
- *Assumptions* Similar to constraints, assumptions are recorded in many ways, from the explicit descriptions of context domains, requirements and architectural solutions, to the articulation of concerns and their risks in the justifications of transformation steps. All assumptions are visible for scrutiny and validation to stakeholders. This can be seen throughout the case study.
- *Suitability* Transformation steps are guarded by justifications in POE and must be discharged to establish the adequacy of the resulting problem to stakeholders. Also any proposed solution must be validated with relevant stakeholders to establish that they are fit for purpose [20], with justifications building the adequacy argument for the resulting solution.
- *Alternatives and trade-offs* Solution exploration may result 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.
- *Traceability* Traceability is a by-product of the POE requirements that explicit problem models, with associated descriptions, be produced, that explicit step justifications be provided and validated, that key stakeholders be identified and associated with validation problems, and that problem solving should proceed in a step-wise manner capturing relationships between problems. Each of these traceability aspects can be observed in the case study. Both POE transformation diagrams and design trees are tools for traceability which were used in the study.
- *Status* Status information is explicitly recorded against each concern expressed in a justification, reflecting the position of each transformation step within the POE process of intertwined exploration and validation activities.
- *Issue descriptions.* Explicit descriptions are required by POE in all exploration and validation activities in order to cover a wide range of design issues of interest to

stakeholders. Examples from the case study, include specific concerns in justifications, such as validity or compliance, and the Triage Document from which the initial model description of the Valid Amounts problem was derived and then validated by the SPST.

Colwell [11] 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. From the study, it transpired that many POE tools help one to deal with design complexity, including the identification and separation of exploration and validation activities, of problem elements and their relationships, and of distinct concerns associated to each design step. Such tools seem to allow the designer to focus at any one time on distinct parts of a problem, thus limiting the design ideas that need to be considered in one go, while still providing ways of piecing all elements of problem solving together, hence addressing the overall complexity of the design.

8.2 Compatibility and improvement of current practice

A particular concern in the study was that the application of principles and techniques should not require practitioners to abandon their tried and tested processes and practices, but should instead strengthen and augment them whenever appropriate. As highlighted in Sect. 7, due to the non-prescriptive nature of POE, it was found that the approach combined well with existing practice within the organisation, and that the extra effort related to performing such an analysis and knowledge capture, rather than being perceived as an intrusive add-on, was found to add value to the process, to a point that this was subsequently modified to include such an analysis and corresponding validation with the Client and their support team as a matter of routine. It was felt that such an addition to the current process would provide a way to mitigate the loss of information related to the transient nature of the communication which was typical of the process prior to this study, addressing some of the concerns expressed by respondents to the initial survey as to not knowing where to find the information, the loss of information held by individuals and the lack of design documentation for reference made available to business analysts.

8.3 Validity of the study

The limitations of a single case study are widely acknowledged [16] and as Rogers observes [40]:

> Successful practice, that is, a successful application of a theory, does not by itself validate the theory.

Nevertheless we can still argue the validity of our results from a number of perspectives.

The first author's in-depth knowledge of the organisational practice, combined with the POE expertise of the co-authors, provides some degree of confidence that the data generated during the study are both credible and dependable: particular care was taken in selecting the process under study and in the generation of all related documentation to a standard which was acceptable to organisational norms and in accordance with the theoretical and principled basis of POE.

As the process under study is fairly typical of software supplying organisations, and complies with industry-wide guidelines on service management, it is also reasonable to conclude that a similar approach could be taken for related processes in other organisations, hence providing some confidence in the transferability of the results. It should also be noted that this research was conducted in the context of a wider programme of POE research where further evidence of transferability has been and will continue to be collected. In particular, the ability to combine POE with existing industrial design processes and practices, and to identify foci for improvement, is something that we have also noticed in other studies, past (e.g., [30]) and current, hence providing further support to our transferability claim.

9 Related work

Despite few decades of research on design rationale, there is little evidence of actual adoption in an organisational settings, therefore the study we have conducted provides a rather unique perspective.

Particularly in the 80s and early 90s, a body of work looked at design rationale in the context of Software Engineering (surveys can be found in [25,39,41]). Much work focused on argumentation models underpinning design rationale (e.g., [27,31,37]), some with accompanying automatic tools (e.g., [12]). Some approaches were based on Toulmin's model of argumentation [42], which has also been used in the assurance world as the basis of safety critical argumentation [4]. Some have been influential in requirements engineering as the basis for requirements elicitation (e.g., [36]). However, by and large, these works are in isolation of the actual process of designing artefacts, and shed no particular light as to how they could be applied in the context of existing organisational processes. In fact, there is very little evidence that they have actually been applied outside academia. Instead our work looks at design rationale capture in a real-world context within existing organisational processes. We also look at a wider range of artefacts and knowledge capture for design traceability [14] beyond argumentation, although arguments do play a key role in our approach too. The type of arguments we focus on are those which are relevant to stakeholder validation, the form of which is likely to be different in different organisational contexts. The form adopted in this

study is based on the notion of 'design concern', which was derived from empirical work we conducted in the context of high assurance system engineering [18], so it has his roots in engineering practice rather than been driven purely by theory.

Work from MacLean et al. [29] shares with our approach the idea that design and design rationale capture should intertwine, and design rationale be an outcome of the design process alongside the designed artefact. To this effect they define a "design space" in which rationale is captured as the combination of a decision space, where decisions and alternatives are recorded, and an evaluation space, where criteria for explaining and understanding such decisions are provided. Such a design space is still decoupled from the actual process of designing the artefact—which is not explicitly considered in the approach—while we see the design of the artefact and a representation of its key properties as key contributors to design rationale capture, with validation artefacts and arguments providing the supporting justifications for design choices. The recognition and involvement of a variety of stakeholders, key to our approach, is also not considered in that work.

Although not specifically on design rationale, a comparison can be made between POE and Soft Systems Methodology [8,9]. Both have their roots in an engineering tradition, and aim at tackling complex real-world issues with particular attention to their social context. There are, however, some key differences. SSM aims at promoting continuous change to ameliorate complex real-world problematic situations through cycles of analysing the situation, taking into consideration and reconciling a wide range of possibly conflicting stakeholders' viewpoints, and agreeing on actions to be taken. As part of the analysis, models are built as 'rich pictures' to help one with understanding of the situation, conflict resolution and agreement on actions. SSM is a highly analytical approach, but provides little support for synthesis: it is unspecific as to how models should be constructed or actions carried out. SSM is also agnostic to the concepts of risk, traceability and design rationale. POE aims at underpinning the engineering and design of fit-for-purpose artefacts by providing both analytical and synthetic tools. As such it is specific in providing structures and mechanisms for both. For instance, it requires that specific knowledge about context and needs be captured, or that specific stakeholders' roles should be enacted. It also use assurance—through identification and mitigation of risk via stakeholders' validation, and traceability of artefacts and decisions—as a driver for design, which is accomplished through a constructive problem solving process.

A parallel can also be drawn between POE and Agile software development methods [1]. The latter emerged as a reaction to the heavy-weight plan-driven software development methods which were particularly popular in the 90s. Agility (the umbrella principles of all the ensuing methods) aimed at identifying better ways of developing software by favouring [2]: *"individuals and interactions over processes and tools; working software over comprehensive documentation; customer collaboration over contract negotiation; responding to change over following a plan"*. On the other hand POE builds on an engineering tradition and aims at better ways of designing fit-for-purpose artifacts (including software) by valuing: stakeholder validation alongside artifact design; traceability and transparency of design decisions; assurance as a driver for design; the use of risk/resources trade-offs to balance design processes. As a consequence, beside POE's wider scope of application beyond software, the two are not incompatible, but have a different bias. From the viewpoint of design rationale capture, POE's mechanisms for assurance already embody much of what is required for design rationale capture, something which is lacking in Agile methods for which the production of working software overshadows any other consideration.

10 Conclusion

This paper has addressed the viability and benefits of capturing design rationale as a by-product of design in the context of a real-world global organisational setting. This was achieved through a study in which POE was applied in the context of a global financial institution to address a specific IT problem as part of its software supplier's client resolution process.

Notwithstanding the limitations of a single case study, the work has provided some positive evidence that POE was instrumental in guiding knowledge capture of key design rationale elements including knowledge of the designed artefacts, the design process, the decisions made during design and their justifications. Specifically, POE provided guidance as to which information should be collected—the 'what' question—through the notion of problem and its constituent parts, and the point at which it should be collected—the 'when' question—through its transformational step-wise approach. Both of the salient views (process-based and artefact-based) and their relations could be captured through POE techniques, including a record of any backtracked decision during development.

Moreover, due to the non-prescriptive nature of POE, it was found that the approach combined well with existing practices within the organisation, which could be exercised as usual, but augmented with POE related activities. It was found that, rather than a burden, the extra effort related to the application of POE principles and techniques had added value to the process, which was subsequently modified to include such activities as a matter of routine.

Following from the study, the following areas have been identified as possible subjects of further research. Further validation that POE can positively address the time and budget limitations that inhibit design rationale capture should be

sought. While our study is an initial step, a more detailed investigation could look into quantifying the potential gain that could be achieved in these areas: for instance, subsequently, the Software Supplier indicated anecdotally that since the introduction of the Problem Specification document they had experienced a reduction in the number of solutions returned for rework, however, a quantification of such a reduction would be beneficial. This would also contribute to promoting design rationale capture, as well as encouraging the uptake of POE, within organisations.

Regli et al. [39] surmise that the way in which design rationale is stored and presented for retrieval is of critical importance to 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.

Lack of tool support for design rationale capture is also identified as a limitation to its uptake [41]. 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 tools 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.

Acknowledgments We are pleased to acknowledge the support of IBM, under the Eclipse Innovation Grants, of SE Validation Ltd., of CWTech Ltd and Bank of America. We thank our colleagues in the Department of Computing at The Open University, particularly Michael Jackson, for their continuing support, and Guy Nkwocha for his patience, support and understanding.

References

1. Agile Alliance. http://www.agilealliance.org/home. Accessed 9 Dec 2011
2. Agile Manifesto. http://agilemanifesto.org/ (2001)
3. Beck, J.: IT services sourcing goes strategic. Gartner, Stamford (2002)
4. Bloomfield, R., Bishop, P., Jones, C., Froome, P.: ASCAD—Adelard Safety Case Development Manual (1998)
5. Burge, J.E., Brown, D.C.: Reasoning with design rationale. In: Proceedings of Artificial Intelligence in Design'00, pp. 611–629. Kluwer, The Netherlands (2000)
6. Burge, J.E., Brown, D.C.: Integrating design rationale with a process model. In: Proceedings of the Workshop on Design Process Modelling, Artificial Intelligence in Design'02, Cambridge, UK (2002)
7. Canas, A., Leake, D., Wilson, D.: Managing, mapping, and manipulating conceptual knowledge. In: Proceedings of the AAAI-99 Workshop on Exploring Synergies of Knowledge Management and Case-Based Reasoning (1999)
8. Checkland, P.: Systems Thinking, System Practice. Wiley, New York (1981)
9. Checkland, P., Poulter, J.: Learning for Action. Wiley, New York (2006)
10. Clements, P.: Comparing the SEI's views and beyond approach for documenting software architectures with ansi-ieee 1471-2000. Technical report, CMU/SEI (2005)
11. Colwell, B.: Complexity in design. IEEE Comput. **38**(10), 10–12 (2005)
12. Conklin, J., Begeman, M.L.: glBIS: a hypertext tool for exploratory policy discussion. ACM Trans. Office Inf. Syst. **6**(4), 303–331 (1988)
13. Cross, N.: Expertise in design: an overview. Des. Stud. **25**(5), 427–441 (2004)
14. Dick, J.: Design traceability. IEEE Software (2005)
15. Douglas, I.: Capturing and managing decision making rationale. In: Proceedings of the 2005 IEEE International Conference on Information Reuse and Integration, pp. 172–176, Las Vegas, Nevada August 15–17 (2005)
16. Flyvbjerg, B.: Case study. In: Denzin, N.K., Lincoln, Y.S. (eds.) The Sage Handbook of Qualitative Research, Chapter 17 (4th edn), pp. 301–316. Sage Publications, London (2011)
17. Gruber, T., Russell, D.: Design knowledge and design rationale: A framework for representing, capture, and use. Technical report, Knowledge Systems Laboratory. Standford University, California, USA (1991)
18. Hall, J.G., Mannering, D., Rapanotti, L.: Arguing safety with problem oriented software engineering. In: Proceedings of the 10th IEEE International Symposium on High Assurance Systems Engineering (HASE 2007), Dallas, Texas, November 14–16 2007. IEEE Computer Society. http://oro.open.ac.uk/9591/
19. Hall, J.G., Rapanotti, L.: Requirements analysis in context with POE Design. In: Proceedings of the International Workshop on Requirements Analysis, London, 6–7 December 2008. Pearson. http://oro.open.ac.uk/19145/
20. Hall, J.G., Rapanotti, L.: Assurance-driven design in problem oriented engineering. Int. J. Adv. Syst. Meas. **2**(1), October, 26–31. http://oro.open.ac.uk/19123/ (2009)
21. Hall, J.G., Rapanotti, L., Jackson, M.: Problem oriented software engineering: A design-theoretic framework for software engineering. In: Proceedings of IEEE Software Engineering Formal Methods (SEFM 2007), London, UK. http://oro.open.ac.uk/9592/ (2007)
22. Hall, J.G., Rapanotti, L., Jackson, M.A.: Problem oriented software engineering: Solving the package router control problem. IEEE Trans. Softw. Eng. **34**(2) (2008). http://oro.open.ac.uk/5445/
23. Jackson, M.A.: Problem Frames: Analyzing and Structuring Software Development Problem, 1st edn. Addison-Wesley Publishing Company, Reading (2001)
24. ITIL: The IT Infrastructure Library: ITIL Home. Office of Government Commerce (OGC). http://www.itil-officialsite.com/ (2010)
25. Jarczyk, A., Loffler, P., Shipman, F.: Design rationale for software engineering: a survey. In: Proceedings of 25th Annual Hawaii International Conference on System Sciences (1992)
26. Kruchten P., Lago P., Van Vliet H., Wolf, T.: Building up and exploiting architectural knowledge. In: Proceedings of the 5th Working IEEE/IFIP Conference on Software Architecture, WICSA 2005, pp. 291–292. IEEE Computer Society (2005)
27. Kunz, W., Rittel, H.: Issues as elements of information systems. Working Paper No. 131, Institute of Urban and Regional Development, University of California, Berkeley, California (1970)
28. Lee, J.: Design rationale systems: understanding the issues. IEEE Expert **12**(3), 78–85 (1997)
29. MacLean, A., Young, R.M., Moran, T.P.: Design rationale: the argument behind the artefact. In: Proceedings of CHI'89, pp. 247–

252, Austin, Texas, April 30–June 4 1989. ACM Press, New York (1989)
30. Mannering, D., Hall, J.G., Rapanotti, L.: Safety process improvement with POSE and Alloy. In: Saglietti, F., Oster, N. (eds.) Safety, Reliability, and Security, Proceedings of the 26th International Conference on Computer Safety, Reliability and Security (Safecomp 2007), Lecture Notes in Computer Science, vol. 4680, pp. 252–257, Nuremberg, Germany, 18–21 September (2007). Springer, Berlin. http://oro.open.ac.uk/9587/
31. McCall, R.J.: Phi: a conceptual foundation for design hypermedia. Des. Stud. **12**(1), 30–41 (1998)
32. Nkwocha, A.: Design Rationale Capture with Problem Oriented Engineering: an Investigation into the Use of the POE Framework for the Capture of Design and Architectural Knoweldge for Reuse within an Organisation. Master's thesis, The Open University (2009)
33. Nonaka, I., Takeuchi, H.T. K.-C.C.: The Knowledge-Creating Company. Oxford University Press, Oxford (1995)
34. Oates, B.J.: Researching Information Systems and Computing. Sage Publications, London (2006)
35. OMG: Business Process Model And Notation (BPMN), version 2.0
36. Potts, C.: ScenIC: a strategy for inquiry-driven requirements determination. In: Proceedings of the IEEE Fourth International Symposium on Requirements Engineering (RE'99), Limerick, Ireland, June 1999. IEEE Computer Press
37. Potts, C., Bruns, G.: Recording the reasons for design decisions. In: Proceedings of the 10th international conference on Software engineering (ICSE '88), Singapore, April 11–15. IEEE Computer Press (1988)
38. Rapanotti, L., Hall, J.G.: Designing an online part-time master of philosophy with problem oriented engineering. In: Proceedings of the Fourth International Conference on Internet and Web Applications and Services, Venice, Italy, May 24–28 2009. IEEE Press. http://oro.open.ac.uk/15331/
39. Regli, W., Hu, X., Atwood, M., Sun, W.: A survey of design rationale systems: approaches, representation, capture and retrieval. Eng. Comput. **16**(3–4), 209–235 (2000)
40. Rogers, G.: The Nature of Engineering. The Macmillan Press Ltd, Basingstoke (1983)
41. Tang, A., Babar, M., Gorton, I., Han, J.: A survey of architecture design rationale. J. Syst. Softw. **79**(12), 1792–1804 (2006)
42. Toulmin, S.: The Uses of Argument. Cambridge University Press, Cambridge (1958)
43. Yin, R.K.: Case Study Research. Design and Mothods, 4th edn. Sage Publications, London (2009)

Author Biographies

A. Nkwocha MSc, is a Prince2 Qualified Technical Project Manager with CWTech Solutions (Bank of America subsidiary). She holds an MSc in Software Development from The Open University, and has 24 years of IT industry experience in programming, analysis and application support and management. Her research in Problem Oriented Engineering has delivered value to her employers in her role in Production Support by improving processes and reducing rework.

J. G. Hall MBA, PhD, CEng, CSci, FBCS CITP, Fellow of IARIA, is a Senior Lecturer and Researcher in the Computing Department at The Open University, UK. His research mission is to provide an overarching theory of engineering design into which the traditional and modern engineering disciplines, such as computing, fit, and within which their extension, interrelatedness and complementarity can be explored, to which this work contributed. He is the Editor in Chief of Expert Systems: The Journal of Knowledge Engineering and of the International Journal of Software Advances, and sits on the editorial board of The International Journal of Learning and Intellectual Capital. He has published around 100 academic papers in areas that include Petri Nets and Process Algebras; Logic; Formal Methods for Safety-Critical Systems; Requirements, Software, Systems and Knowledge Engineering; Design; Artificial Intelligence and Multi-Agent Systems; he has also edited many special issues and books.

L. Rapanotti PhD, FBCS, is a Senior Lecturer in the Computing Department, The Open University, UK, graduated with a Laurea Cum Laude (Milan, Italy) and a PhD (Newcastle upon Tyne, UK) in Computer Science. Lucia is a past Editor in Chief and current Editorial Board Member of Expert Systems: The Journal of Knowledge Engineering and is a Fellow of BCS, The Chartered Institute of IT. Previously, she held research positions at Milan University (Italy), Oxford University and the University of Newcastle upon Tyne (UK). Lucia's research focuses on the development of theories and tools to help people tackle complex design problems, whether in the development of software, systems, processes or other artifacts. She has published widely in international conferences and journals and is a member of numerous international programme committees.