ISSN 1744-1986

Technical Report N^o 2008/03

Capturing Knowledge through Problem Oriented Engineering

Jon G. Hall
Lucia Rapanotti

19th March 2008

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

http://computing.open.ac.uk

Capturing Knowledge through Problem Oriented Engineering

Jon G. Hall Lucia Rapanotti
Computing Department
The Open University, UK
{J.G.Hall, L.Rapanotti}@open.ac.uk

Abstract

Problem Oriented Engineering (POE) is a formal system for engineering design. In previous work, we have successfully applied POE within the context of software engineering. This paper illustrates the application of POE beyond software to capturing knowledge in a socio-technical context. The problem we are considering is that of publishing. Through a POE analysis we capture the essential elements of the problem and show how scientific journals provide a tried and tested solution to this problem, exposing the rationale behind their success.

1 Introduction

Problem Oriented Engineering (POE) is a formal system for engineering design.

Engineering design (shortly, design) is the creative, iterative and often open-ended endeavour of conceiving and developing products, systems and processes (adapted from [ECS03]). By necessity, it includes the identification and clarification of requirements, the understanding and structuring of the context into which the designed artefact will be deployed, the specification of a design for a solution that can ensure satisfaction of the requirements in context, and the construction of arguments, convincing for all validating stake-holders, that the designed artefact will provide the functionality and qualities that are needed. The involvement of stake-holders motivates collecting and recording evidence of the designed artefact's fitness for purpose.

In previous work [HMR07, MHR07c, MHR07b, MHR07a], we have shown how the Problem Oriented Engineering (POE) framework, instantiated as Problem Oriented Software Engineering (POSE) [HRJ08], can be used to support engineering design of software.

Recently, we were asked to give an account of scientific publishing to an academic audience interested in improving their academic stand through journal publication. This made us reflect on the relation between the publishing industry and academia and the particularly chemistry which makes such a relation work so effectively. We used POE as a tool for capturing knowledge and what follows is an account of our analysis.

The paper is structured as follow. Section 2 provides a brief introduction to POE. Section 3 discusses POE for capturing knowledge and details its application to publishing. Section 4 reflects on what has been achieved in the paper and offers some conclusions.

2 Problem Oriented Engineering

Problem Oriented Engineering (POE)[1] is a Gentzen-style natural framework for engineering design (see, for instance, [Pel99]). As such, POE supports rather than guides its user as to the particular sequence of design steps that will be used; the user choosing the sequence of steps that they deem most appropriate to the context of application. The basis of POE is the *problem sequent* for representing *design problems* requiring designed solutions. The transformations defined in POE transform problems as sequents into others in ways that preserve solutions (in a sense that will become clear). When we have managed to transform a problem to axioms[2] we have solved the problem, and we will have a designed solution for our efforts.

POE is a formal system for working with non-formal and formal descriptions, as POE is designed to work with problems not propositions as in the original natural deduction: the characteristic that distinguishes it most from natural deduction is the guarding of transformations by *justification obligations*, the discharge of which establishes the 'soundness' of the application *with respect to stake-holders*. Natural deduction is based on a single absolute notion of correctness provided by proof whereas, through justifica-

[1] A full presentation of the POE framework is beyond the scope of this paper but can be found, instantiated for software design, in [HRJ08].

[2] An *axiomatic problem* is a problem whose known fit-for-purpose solution is known.

tions, POE caters for the engineering notion of fitness-for-purpose, something that is often very far from correctness.

A *POE problem* has three elements: a real-world context, W, a requirement, R, and a solution, S. The problem context is a collection of *domains* ($W = D_1, ..., D_n$) described in terms of their known, or *indicative*, properties, which interact through their sharing of *phenomena* (i.e, events, commands, states, *etc*. [Jac01]). The problem requirement states how a proposed solution description will be assessed as the solution to that problem. Like a domain, a requirement is a named description with phenomena; a requirement description should always be interpreted in the optative mood, i.e., as expressing a wish. A solution is a domain, intended to solve a problem, i.e., when introduced into the problem context will satisfy the problem requirement. The possible descriptions of a solution range over many forms, from high-level specification through to detailed designs.

A problem's elements come together in POE in a *problem sequent*:

$$D_1, \ldots, D_n, S \vdash R$$

Here $\vdash$ is the *problem builder* and reminds us that it is the relation of the solution to its context and to the requirements that we seek to explore. By convention, the problem's solution domain, S, is always positioned immediately to the left of the $\vdash$.

One way to visualise a POE problem is illustrated in Figure 1[3], where the problem is to design a solution for the safe operation of a two-button press, a medium-size press commonly used in the machine tool industry, usually working in close proximity to a human operator. The operator puts some material to be pressed on the press table, and pushes two buttons that order the press to activate. The two buttons are used for safety: the operator needs to use both hands to press the buttons at the same time (one for each button) and, as a consequence, the risk of hand injury is reduced greatly.

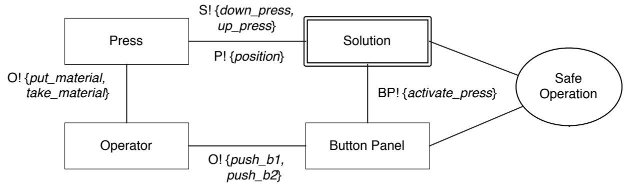

Figure 1. The 2-button Press Problem

In the diagram, the undecorated rectangles, Operator, Button Panel and Press are context domains; the decorated rectangle, Solution, is the solution to be found; the oval, Safe Operation is the requirement; and the arc annotations are shared phenomena. We will adopt this notation throughout the paper for illustration of the various formal POE problem definitions we will encounter.

The descriptions of a problem's elements may be in any language, different elements being described in different languages, should that be appropriate. So that descriptions in many languages may be used together in the same problem, POE provides a *semantic meta-level* for the combination of descriptions; notationally, this is a role of the ',' that collects into a problem sequent the domains that appear around the turnstile, formally making each visible to the others through their shared phenomena[4]. Throughout the paper we will use natural language descriptions of problem elements.

POE problem transformations capture discrete steps in the problem solving process, by relating a problem and a justification to (a set of) problems. Problem transformations conform to the following general pattern. Suppose we have problems $W, S \vdash R$, $W_i, S_i \vdash R_i$, $i = 1, ..., n$, ($n \geq 0$) and justification J, then we will write:

$$\frac{W_1, S_1 \vdash R_1 \qquad \ldots \qquad W_n, S_n \vdash R_n}{W, S \vdash R} \begin{smallmatrix}[\textrm{NAME}] \\ \langle\langle \textrm{J} \rangle\rangle\end{smallmatrix}$$

to mean that, derived from an application of the NAME problem transformation schema (discussed below):

> S is a solution of $W, S \vdash R$ with *adequacy argument* $(CA_1 \wedge ... \wedge CA_n) \wedge J$ whenever $S_1, ..., S_n$ are solutions of $W_1, S_1 \vdash R_1, ..., W_n, S_n \vdash R_n$, with adequacy arguments $CA_1, ..., CA_n$, respectively.

By specialising this pattern, many classes of transformations are recognised in POE (see [HRJ08]), reflecting a variety of engineering practices reported in the literature or observed elsewhere. Each of them prescribes the way in which the *conclusion* problem (that below the line) is related to the *premise* problem(s) (those above the line), and which form the justification must take, called *justification obligation*. When applying a pattern, the justification must be discharged by providing evidence which contributes towards the overall adequacy argument.

2.1 Justification general form

POE explicit requirement for discharging justification obligations at each step is a distinguishing feature of the framework. From our experience of repeated application of POE to engineering problems, the following general form of the justification has emerged:

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

[3] The notation is reminiscent of that of Problem Frames [Jac01], which shares a similar notion of problem as POE, albeit aimed at software specification.

[4] A situation similar to that found in the propositional calculus in which conjunction and disjunction, *etc*, serve to combine the truth values of the atomic propositions.

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

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

CONCERN: Name
STATUS: Status

A concern (c.f., [Jac01]) is something that is important to the development, presumably because it relates to some stake-holder in the process. In high integrity development, for instance, the reliability concern is likely to arise; a design that does not address such a concern in such a context is likely to be unvalidatable. The status of a concern is one of *pending, discharged, undischargeable*. The work appertaining to the discharge of a concern is structured: each concern has associated with it the following:

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

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

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

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

CONCERN: Step Validity
STATUS: Status

The status of the step validity concern, possible values include *pending, signed-off, undischargeable*

ARGUMENT & EVIDENCE: Explanation of the status after validation, including the relationships where evidence was gathered in the design, and the treatment chosen for the residual risk of the step.

SIGNATORY: To recognise the stake-holder or stake-holders that signed-off the step.

Each element is optional, typically depending on the developmental stage and context. Similarly, the collection of concerns at each step application depends on the particular transformation pattern applied.

A concern leads to a *claim* stated within a justification, the claim being that the concern is discharged by the development step. The justification will, eventually, contain arguments and evidence that the claim is valid so that the concerns is discharged. We say eventually because some concerns can only be discharged after the ramifications of a problem transformation are known which is, typically, later in the development tree.

One particularly important concern is the *step validity concern*—for which the associated claim is that a particular step is validatable—as it is the point of contact of the POE process with stake-holders external to the creative process of the problem solver, as explained below.

2.2 Process pattern

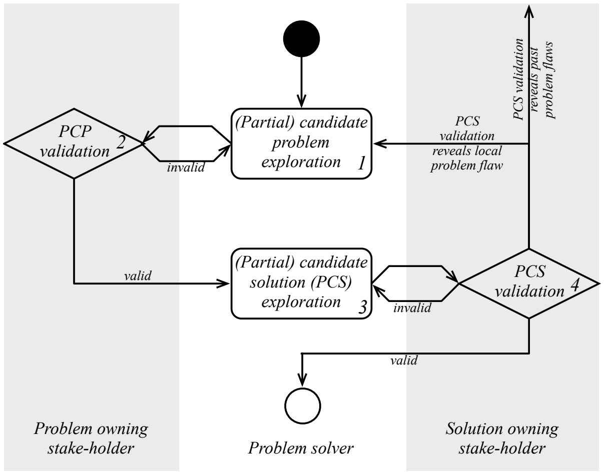
PCS validation reveals past problem flaws
(Partial) candidate problem exploration 1
PCP validation 2
invalid
PCS validation reveals local problem flaw
valid
(Partial) candidate solution (PCS) exploration 3
PCS validation 4
invalid
valid
Problem owning stake-holder
Problem solver
Solution owning stake-holder

Figure 2. POE process Pattern: to move towards a partial solution to a general engineering problem, we first understand the problem better (1), reflecting our understanding of the problem through validation with the problem holding stake-holder (2); use engineering judgement to determine a candidate solution architecture (3), then test the candidate for satisfaction of concerns, iterating if necessary (4).

Typically, engineering design under POE proceeds through repeated application of the process pattern of Figure 2, captured as a UML activity diagram [OMG]. The pattern highlights three main constituents—activities, choice points and roles—which we will discuss in turn the following. Two main activities are present:

(Partial) Candidate Problem Exploration: to capture (increasing) knowledge and detail in the context and requirement of the problem;

(Partial) Candidate Solution Exploration: to structure the solution (or part thereof) according to a candidate architecture.

The partial nature of the candidates is so that early problem solving can focus on parts of the problem or solution, rather than the whole problem straight away.

The choice points are:

(Partial) Candidate Problem Validation: to determine whether the current candidate characterisation of the problem context and requirement is appropriate to start investigating a solution, or we need explore the problem further;

(Partial) Candidate Solution Validation: to determine whether the current candidate solution is viable as the basis of a solution or whether, instead, we should backtrack the development to find another candidate solution or explore the problem further.

The role are those of *problem owning stake-holder(s)*, *solution owning stake-holder(s)*, and *problem solver*, their respective scopes indicated by shading in the figure (note that the roles, as such, do not overlap).

A *problem owning stake-holder* is someone who role is to validate a (parital) candidate problem description that results from Partial Candidate Problem Exploration. There are many familiar examples of problem owning stake-holders. These include, but are not limited to, those of customer (those that pay for a product), clients (those that pay for a service), regulator (those requiring safety, for instance), end-user (those who will use the product or service when commissioned). It is the problem owning stake-holders' role to answer the question "Is this (partial) problem description valid for you?" Depending on the problem-owning stake-holders' responses, the problem solver may need to re-explore the problem (when the answer is "No!"), or move on to try to find a (partial) solution (when the answer is "Yes").

The role of the *solution owning stake-holder(s)* is to validate a candidate solution description, such as an architecture (a partial solution) or choice of component (i.e., something of complete functionality). Solution owning stake-holders include, but are not limited to, a development house's chief software architect—who knows which architectures their organisation uses in solutions, an oracle—who determines which of a number of features should be included in the next release, or a project manager—who needs to timebox particular activities; there are many other roles that fit solution owning stake-holder. It is the solution owning stake-holders' role to answer the question "Is this (partial) solution description valid?" Depending on their response, the problem solver may need to re-explore the solution (when the answer is "No!"), move back to exploring this or a previous problem (when the answer is "No, but it throws new light on the problem!"), or moving on to the next problem stage (when the answer is "Yes!").

The role of *problem solver* is that of the person(s) that begins by trying to understand the problem and iterates towards a solution. As indicated by the upward pointing arrow that appears in the upper right of Figure 2, iteration is not always local: it is, for instance, possible that through the failed validation of a solution a previous problem description may be revealed as flawed, even if it has been validated by a problem-owning stake-holder and so invalid—problem-owning stake-holders make mistakes too!

It is worth emphasising that we do not preclude communication between those that will perform the role of problem- or solution-owning stake-holder, or problem solver during the process of problem solving. Indeed, this would be a very sensible option—even if just to manage the expectations of the various stake-holders before the formal validation is conducted.

Note that the step validity concern associated with a problem exploration step is dischargeable only with reference to the problem-owning stake-holder. Similarly, the step validity concern associated with a solution exploration step is dischargeable only with reference to the solution-owning stake-holder. It is the discharge of step validity concerns that *require* the problem solver to consult with stake-holders (although, of course, consultation with stake-holders may also take place in problem and/or solution exploration).

On the other hand, like other concerns, the discharge of step validity concerns may be postponed. Depending on the criticality of a development, the risk exposed by such a postponement may be unacceptable—given that a problem- or solution-owning stake-holder has not validated a partial problem or solution candidate, the problem solver may be solving the wrong problem with incorrect solution technologies, or both. In this case, the future development is based on an assumption of validity. The commitment of developmental resources on this assumption is the source of the risk, although it may be more or less mitigated by problem solver experience. Of course, even if the risk is managed by discharging the step validity concern, there may be secondary risks, such as the a problem-owning stake-holder being incorrect in their validation. It may therefore be important, as part of the justification for the development step to record the explicit instance of step validity concern discharge so that it is traceable; the recording of concern discharges are properly a part of all POE steps.

3 POE for capturing knowledge

It could be argued that POE problem exploration is, in fact, a process of capturing knowledge, aimed at producing a model which encapsulates knowledge of the real world context, and of stakeholders' needs. Such a model is then used to synthesise an appropriate solution.

It is plausible, however, to regard solution exploration also as capturing knowledge, as solutions are not always the outcome of radical design [Vin90], but are often the result of reusing and adapting tried and tested knowledge: in such cases, uncovering what already exists is an essential part of solution exploration. This is particularly true in change scenarios, in which shortcomings of current systems need to be identified as the basis for future improvement.

In this section we report on the use of POE as an analytical framework for capturing knowledge within the context of scientific publishing. In the development that follows, we apply the POE process pattern, with validating stakeholders members of the editorial team of Expert Systems, a scientific journal on Knowledge Engineering [Exp] of which the authors are co-Editors in Chief.

Publishing is a process by which information is produced and disseminated. Although publishing could be traced back to the ancient world, publishing as a mass industry started with the invention of the printing press in 15*th* century, and has grown steadily since to the global enterprise we experience today. Of the many aspects of modern publishing, we will focus on the contribution made by scientific journals and uncover the rationale behind their success.

3.1 Publishing as a POE problem

Our initial problem exploration led to the POE characterisation of publishing illustrated in Figure 3. The context includes one domain, the Readership domain (undecorated rectangle at the bottom of the figure), indicating the readership of marketed publications. The object of knowledge capture is the Publishing Industry (decorated rectangle at the top in the figure). The requirement *Publishing* (oval to the right) is what we observe to be true, that is the mass communication of information through the exchanged of publications and fees between industry and readership. The phenomena annotations on links indicate that fees are paid by the Readership domain to the Publishing Industry in exchange for publications.

Formally in POE:

STEP ID*: Application of* CONTEXT AND REQUIREMENT EXPLORATION *as a starting point*

JUSTIFICATION J_1: An initial characterisation of the problem is:

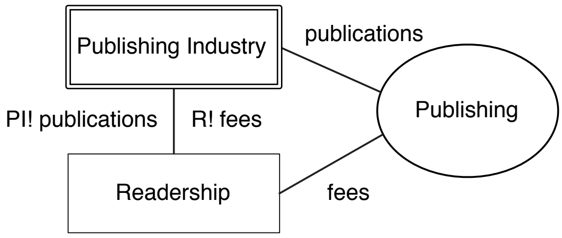

Figure 3. The Publishing Problem

$$P_1 : \begin{array}{l} \mathit{Readership}^{\mathit{fees}}_{\mathit{publications}}, \mathit{Publishing\ Industry}^{\mathit{publications}}_{\mathit{fees}} \\ \vdash \mathit{Publishing}^{\mathit{fees}}_{\mathit{publications}} \end{array}$$

DESCRIPTIONS & PHENOMENA: Here are the initial context and requirement descriptions:

Publishing Industry	Industry whose main business goal is mass production and distribution of information.
Readership	Customer base for the publishing industry.
Publishing	Mass communication of information through the exchanged of publications and fees between industry and readership.

and here are their phenomena:

publications	Products marketed by the industry and purchased by the readership
fees	what readership pays for access to publications

CONCERN: Validity
STATUS: Discharged

CLAIM: This is a valid initial characterisation of the problem.

ARGUMENT & EVIDENCE: Historical references and expert opinion were used to validate this initial description.

RISKS: Low

CONCERN: Step Validity
STATUS: Discharged

All step concerns are discharged.

3.2 The Sourcing problem

The publishing industry is made up of publishers who compete in selling their publications to the readership, and have developed as businesses able to provide all necessary printing, marketing and distribution functions, while relying on external sources for the provision of publishable material that they can turn into a product. This is illustrated in Figure 4.

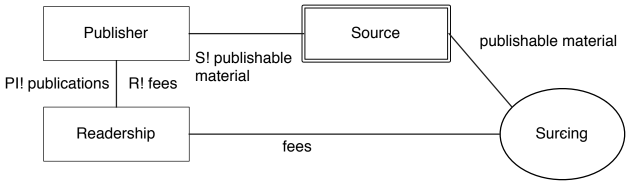

Figure 4. The Sourcing Problem

In POE terms, we have provided an 'architectural structure' for the publishing industry domain in the previous problem, which consists of two components, a Publisher and a Source, where the problem is that for the Publisher to identify a sustainable source of publishable material. The decoration in the figure indicates that Source is the subject of further knowledge capture.

Formally in POE:

STEP ID: *Application of* SOLUTION EXPLORATION *to* P_1

JUSTIFICATION J_2: We choose the following architecture for *Publishing Industry*:

$$PIArch: \quad [Publisher_{publishable\ material}](Source^{publishable\ material})$$

in which *Publisher* is a known component, while *Source* is subject of further investigation. This leads to the following problem:

$$P_2: \begin{array}{l} Readership^{fees}_{publications}, Publisher^{publications}_{fees, publishable\ material}, \\ \quad Source^{publishable\ material} \\ \vdash Sourcing^{fees}_{publishable\ material} \end{array}$$

DESCRIPTIONS & PHENOMENA: Here are the new descriptions:

Publisher	A company whose main business is to sell publication to its readership. It directly implements printing, marketing and distribution functions, while it outsources the provision of publishable material to make into a saleable product.
Source	A source of publishable material the publisher can turned into publications.
Sourcing	Sustainable source of publishable material.

and related new phenomenon:

publishable material	material the publisher can turned into publications

CONCERN: Feasibility
STATUS: Discharged

CLAIM: This is a feasible solution architecture for the *Publishing Industry*.

ARGUMENT & EVIDENCE: Historical references and expert opinion were used to validate the *PIArch* architecture.

RISKS: Low

CONCERN: Step Validity
STATUS: Discharged

All step concerns are discharged.

3.3 Scientific Journals

Scientific journals are known solutions to the publisher problem. A scientific journal usually operates through an editorial board to whom the publisher delegates the responsibility to supply publishable articles. Editorial boards interacts with authors who submit articles for possible publication, and reviewers who assess the quality of the articles on behalf of the editorial board. This is illustrated in Figure 5.

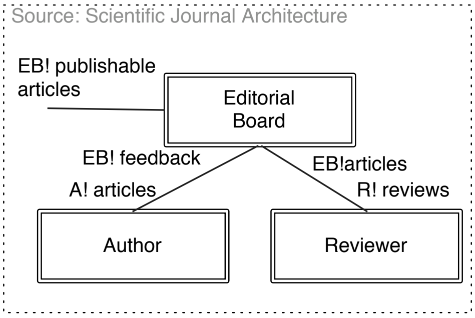

Figure 5. Scientific Journal architecture

What makes this model works as a sustainable source for the publisher is *esteem*. This is an important currency within academic (and, to some extent, professional practitioner) communities, where levels of esteem are attributed to their members based on a collection of indicators, including publishing in reputable scientific journals or being associated with their editorial boards on an ongoing basis. Vice versa, a journal's reputation depends on the quality of their published articles, and is evidenced by its longevity, inclusion in citation indices, impact factor, volume of article submissions, acceptance rate, volume of subscriptions and/or article downloads. It follows that, from an academic perspective, authoring and reviewing for scientific journals, and generally being associated with their editorial boards increases academic esteem, hence many academics will willingly volunteer for such tasks; moreover, editorial boards will strive to improve their journals' reputation and will work hard to attract quality submissions. Naturally, reputation increases readership, which will benefits both academics and publisher.

The formal step is:

STEP ID: *Application of* SOLUTION EXPLORATION *to* P_2

JUSTIFICATION J_3: We choose the following architecture for *Source* in P_2:

$$SArch : \quad [](\mathit{Editorial\ Board}^{\mathit{publishable\ articles,feedback,articles\ for\ review}}_{\mathit{articles,reviews}}, \mathit{Author}^{\mathit{articles}}_{\mathit{feedback}}, \mathit{Reviewer}^{\mathit{reviews}}_{\mathit{articles\ for\ review}})$$

All components of this architecture are object of further knowledge capture, leading to the following three sub-problems (illustrated in Figures 6, 7 and 8, respectively):

$$P_{3,1} : \quad \begin{array}{l} \mathit{Readership}^{\mathit{fees}}_{\mathit{publications}}, \mathit{Publisher}^{\mathit{publications}}_{\mathit{fees,publishable\ articles}}, \\ \quad \mathit{Editorial\ Board}^{\mathit{publishable\ articles,feedback,articles\ for\ review}}_{\mathit{articles,reviews}}, \\ \quad \mathit{Reviewer}^{\mathit{reviews}}_{\mathit{articles\ for\ review}}, \mathit{Author}^{\mathit{articles}}_{\mathit{feedback}} \\ \vdash \mathit{Authoring}^{\mathit{publishable\ articles}}_{\mathit{articles}} \end{array}$$

$$P_{3,2} : \quad \begin{array}{l} \mathit{Readership}^{\mathit{fees}}_{\mathit{publications}}, \mathit{Publisher}^{\mathit{publications}}_{\mathit{fees,publishable\ articles}}, \\ \quad \mathit{Editorial\ Board}^{\mathit{publishable\ articles,feedback,articles\ for\ review}}_{\mathit{articles,reviews}}, \\ \quad \mathit{Author}^{\mathit{articles}}_{\mathit{feedback}}, \mathit{Reviewer}^{\mathit{reviews}}_{\mathit{articles\ for\ review}} \\ \vdash \mathit{Reviewing}^{\mathit{publishable\ articles}}_{\mathit{reviews}} \end{array}$$

$$P_{3,3} : \quad \begin{array}{l} \mathit{Readership}^{\mathit{fees}}_{\mathit{publications}}, \mathit{Publisher}^{\mathit{publications}}_{\mathit{fees,publishable\ articles}}, \\ \quad \mathit{Reviewer}^{\mathit{reviews}}_{\mathit{articles\ for\ review}}, \mathit{Author}^{\mathit{articles}}_{\mathit{feedback}}, \\ \quad \mathit{Editorial\ Board}^{\mathit{publishable\ articles,feedback,articles\ for\ review}}_{\mathit{articles,reviews}} \\ \vdash \mathit{Editorial\ management}^{\mathit{fees}}_{\mathit{publishable\ articles}} \end{array}$$

DESCRIPTIONS & PHENOMENA: The new descriptions are:

Editorial Board	Body responsible for supplying publishable scientific articles to the publisher.
Author	Academic or practitioner submitting articles for publication in a scientific journal.
Reviewer	Academic or practitioner peer reviewing articles for publication in a scientific journal.
Authoring	Maximise author's chances of having their articles published.
Reviewing	Maximise reviewer's chances of a long-term relationship with an editorial board.
Editorial management	Maximise editorial board's chances of a long-term relationship with publisher.

with phenomena:

publishable articles	articles the publisher can publish in scientific journal
articles	authors' articles for consideration by editorial board
reviews	reviews of submitted articles
feedback	feedback from editorial board to authors of submitted articles

CONCERN: Feasibility
STATUS: Discharged

CLAIM: This is a feasible solution architecture for *Source*.

ARGUMENT & EVIDENCE: Scientific journals are an essential tool for academic dissemination, hence provide a business opportunity. Thanks to economy-of-scale and technological advances the publisher can produce, market and distribute journals relatively cheaply. As editorial boards are largely unpaid, the cost of sourcing publishable material is negligible. The publisher assumes the business risk of not having a sufficient level of readership to recover costs and make a profit. This is mitigated by a journal's stand in the community, as witnessed by journal longevity, inclusion in citation indices, impact factors, volumes of article submissions, acceptance rate, volume of subscriptions and/or article downloads, etc. The publisher also assumes the business risk of not having editorial boards able to assure a stream of publishable material. This risk is mitigated by the fact that membership of an editorial board is considered of high esteem in the academic community, hence potential members abound. This argument was validated by experts.

RISKS: Low.

CONCERN: Step Validity
STATUS: Discharged

All step concerns are discharged.

Let us consider the three sub-problem in turn in the following.

3.3.1 Authors

The author sub-problem $P_{3,1}$ (see also Figure 6) is about discovering which author's behaviour will maximise their chances of having their work published.

This problem can be solved by identifying the necessary author's skills and competencies. To this end, we introduce the following acronym, CRISP, which captures essential quality of a scientific article:

C *C*ontribution must be novel and substantial;

R *R*eferences to related work must be included appropriately;

I Must be if *I*nterest to the journal's readership;

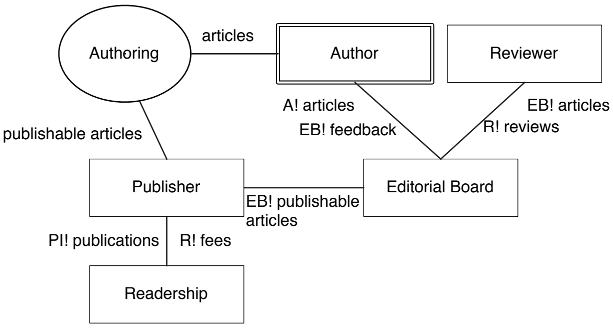

Figure 6. The Author sub-problem

S Technical content must be *S*ound;

P *P*resentation must reflect high quality of narrative and use of language.

Formally:

STEP ID*:* *Application of* SOLUTION INTERPRETATION *to* $P_{3,1}$

JUSTIFICATION $J_{3,1}$:

DESCRIPTIONS & PHENOMENA: A successful author will have the skills and competencies highlighted in the following table (where *Network* refers to the level of access within the academic community which is necessary to fulfil the role[5]):

Role	*Competencies*	*Network*	*Professionalism*
Author	Able to write CRISP articles	n/a	Organised, punctual, ehical, polite and understanding in all interactions with editorial board and reviewers

CONCERN: Feasibility
STATUS: Discharged

CLAIM: This is a feasible solution for *Author*.

ARGUMENT & EVIDENCE: Articles are the main vehicle for academics to make their work known in the community, and volume and quality of publications increase esteem. Also, journals need quality publications to sustain and increase readership, and to improve their impact factors. An author's main risk is paper rejection, which is mitigated by authoring to CRISP.

RISKS: Low.

CONCERN: Step Validity
STATUS: Discharged

All step concerns are discharged.

3.3.2 Reviewers

The reviewer sub-problem $P_{3.2}$ (see also Figure 7) is about discovering which reviewer's behaviour will maximise their changes of a long-term relationship with an editorial board.

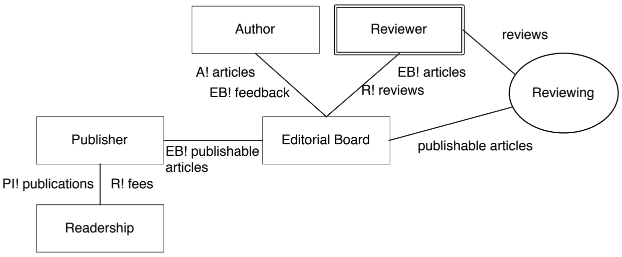

Figure 7. The Reviewer sub-problem

Following a similar development as for the author sub-problem, we arrive at:

STEP ID*:* *Application of* SOLUTION INTERPRETATION *to* $P_{3,2}$

JUSTIFICATION $J_{3,2}$:

DESCRIPTIONS & PHENOMENA: A successful reviewer will have the following skills and competencies:

Role	*Competencies*	*Network*	*Professionalism*
Reviewer	Able to assess CRISP articles	n/a	Organised, punctual, ehical, polite and understanding in all interactions with editorial board and authors

CONCERN: Feasibility
STATUS: Discharged

CLAIM: This is a feasible solution for *Reviewer*.

ARGUMENT & EVIDENCE: Being a reviewer for an international journal increases esteem. Also, journals need expert reviewers to assess quality of paper for publication. Inability to produce informative reviews is mitigated by assessing to CRISP.

RISKS: Low.

[5] A great deal of an academic's work, and in particular research, could not be carried out without access to an international network of other academics.

CONCERN: Step Validity
STATUS: Discharged

All step concerns are discharged.

3.3.3 Editorial Board

The editorial board sub-problem $P_{3.3}$ (see also Figure 8) is about identifying a body whose behaviour will maximise their chances of a long-term relationship with a publisher.

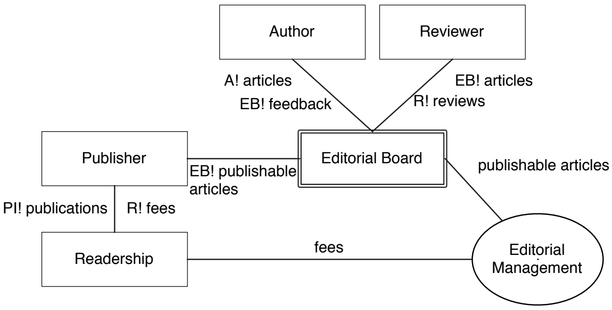

Figure 8. The Editorial Board sub-problem

There are many variants of editorial boards and the way they operate, although they all have very similar processes: they all receive article submissions from authors within a specific scientific discipline, which characterises the journal's content, and engage peer reviewers to assess whether the articles are publishable. The way Expert Systems operates is captured in Figure 9 as solution to the above problem.

The Expert Systems editorial board has four main recognised roles: Editor-in-Chief (EiC), with overall responsibility for the journal and who liaises with the publisher; Associate Editor (AE), who deals with article submissions and their review on a day-to-day basis, and makes recommendation to the EiC on the outcome of the review process of individual articles; Adviser, who generally advises and supports the EiC, performs various editorial tasks at the discretion of the EiC, and generally is an ambassador for the journal in the community; Guest Editor (GE), who deals with article submissions for special issues on a particular topic within the journal's scope. An automatic system, called Manuscript Central (MC in the figure), is used to support all workflows for submissions and reviews of articles.

Formally:

STEP ID: *Application of* SOLUTION INTERPRETATION *to* $P_{3,3}$

JUSTIFICATION $J_{3,3}$:

DESCRIPTIONS & PHENOMENA: Here are the descriptions:

Editor in Chief (EiC)	Has overall responsibility for the journal and for liaising with the journal's publisher.
Associate Editor (AE)	Deal with article submissions and their review on a day-to-day basis; makes recommendation to the EiC on the outcome of the review process of individual articles.
Guest Editor (GE)	Deals with article submissions and reviews for special issues.
Adviser	Generally advises and supports the EiC, performs various editorial tasks at the discretion of the EiC, and generally is an ambassador for the journal in the community
Manuscript Central (MC) system	Supports all workflows for submissions and reviews of articles.

and related phenomena:

R assignment	assignment of a reviewer to a submitted article
recommendations	associate or guest editor recommendation to EiC based on received reviews of a submitted article
submitted articles	all articles submitted for publication in the journal
reviewed articles	all articles who have been reviewed with their reviews and recommendations
AE assignment	assignment of an associate editor to an article in need of reviewing
GE assignment	assignment of a guest editor to an article submitted for a special issue
verdict	EiC verdict on a reviewed article
advice	

A successful editorial board will have the following distribution of skills and competencies:

Role	*Competencies*	*Network*	*Professionalism*
Editor in Chief	Jack of all trades, able to consult, carry out strategic plans, and interact with publisher	Pool of expert authors, reviewers and potential EB members	Organised, punctual, ethical, polite and understanding in all interactions with publisher, editorial board, guest editors, authors and reviewers
Associate Editor	Sustainably able to exercise judgement based on reviewers' comments and make recommendations to EiC	Pool of expert reviewers	Organised and punctual. Ethical, polite and understanding in all interactions with editorial board, authors and reviewers
Guest Editor	Able to exercise judgement based on reviewers' comments	Pool of expert authors and reviewers	Organised, punctual, ehical, polite and understanding in all interactions with editorial board, authors and reviewers
Adviser	Able to advise EiC and to act as ambassador for journal	Pool of expert authors and reviewers	Ethical, polite and supportive in all interactions with EiC

CONCERN: Feasibility
STATUS: Discharged

CLAIM: This is a feasible solution for *Editorial Board*.

ARGUMENT & EVIDENCE: The three-tier structure of the board works well for medium volume of submissions—Expert Systems currently receives just over 100 article submissions per year, plus at least two special issues per volume. From the EiC's perspective this is an efficient structure as a great part of the review process is delegated to AEs and GEs, leaving the EiC the time to fulfil more strategic roles in order to maintain and improve the journal's stand in the academic community. 2008 will be Expert Systems' 25th anniversary, hence the journal has an established reputation and readership. The major risk is loss of reputation though inability to keep publishing relevant, high quality work. This is mitigated through frequent renewal of the editorial board membership and continuous process improvement (e.g., shorting article turnaround, automatic manuscript handling systems, etc.).

RISKS: Low.

CONCERN: Step Validity
STATUS: Discharged

All step concerns are discharged.

This concludes our analysis as the development has now reached a point in which no domain is left for which further knowledge capture is required, and all steps have been fully validated. A summary of the development is given in Figure 10, where labelled circles represent problems, arrows represent problem transformations, and their labels represent step justifications. Checked justifications are validated and leaf problems are solved.

4 Conclusion

This paper has illustrated the application of POE for knowledge capture. Originally conceived for software engineering, POE has developed into a more general framework for engineering design. The POE notion of problem requires a separation of context, requirement and solution, with explicit descriptions of what is given, what is required and what is object of design or knowledge discovery. This improves the traceability of artefacts and their relation, as well as exposing all assumptions to scrutiny and validation. That all descriptions are generated through problem transformation forces the inclusion of an explicit justification that such assumptions are realistic and reasonable.

The example in this paper is from the world of scientific publishing and adds to the growing collection POE examples and case studies.

Acknowledgments

We acknowledge the financial support of IBM, under the Eclipse Innovation Grants, and of SE Validation Limited, in particular Colin Brain for his many comments and insights. We also thank Derek Mannering whose work first instantiated the POE process pattern. Thanks also go to our colleagues in Computing Department at The Open University, particularly Michael Jackson.

References

[ECS03] ECSA. Definition of terms to support the ecsa standards and procedures system. Technical Report G-04, draft 6, Engineering Council of South Africa, Standards and Procedures System, January 2003.

[Exp] Expert Systems, The Journal of Knowledge Engineering. http://www.blackwellpublishing.com/journal.asp?ref=0266-4720.

[HMR07] Jon G. Hall, Derek Mannering, and Lucia Rapanotti. 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.

[HRJ08] Jon G. Hall, Lucia Rapanotti, and Michael A. Jackson. Problem oriented software engineering: Solving the package router control problem. *IEEE Transactions on Software Engineering*, 34(2), March/April 2008.

[Jac01] Michael A. Jackson. *Problem Frames: Analyzing and Structuring Software Development Problem*. Addison-Wesley Publishing Company, 1st edition, 2001.

[MHR07a] Derek Mannering, Jon G. Hall, and Lucia Rapanotti. A problem-oriented approach to normal design for safety-critical systems. In *European Joint Conferences on Theory and Practice of Software (ETAPS)*, Braga (Portugal), 24 March - 1 April 2007. Proceedings of Fundamental Approaches to Software Engineering (FASE) '07.

[MHR07b] Derek Mannering, Jon G. Hall, and Lucia Rapanotti. Safety process improvement: Early analysis and justification. In *Proceedings of the IET Second International Conference on System Safety 2007*, London, UK, 22 -24 October 2007. IET.

[MHR07c] Derek Mannering, Jon G. Hall, and Lucia Rapanotti. Safety process improvement with pose and alloy. In *Proceedings of The 26th International Conference on Computer Safety, Reliability and Security (Safecomp 2007)*, Nuremberg, Germany, 18 - 21 September 2007. Springer.

[OMG] OMG. Unified Modeling Language (UML), version 2.0. http://www.omg.org/technology/documents/formal/uml.htm.

[Pel99] Francis Jeffry Pelletier. A brief history of natural deduction. *History and Phylosophy of Logic*, 20:1–31, 1999.

[Vin90] Walter G. Vincenti. *What Engineers Know and how they know it: Analytical studies from Aeronautical History*. The Johns Hopkins University Press, 1990.

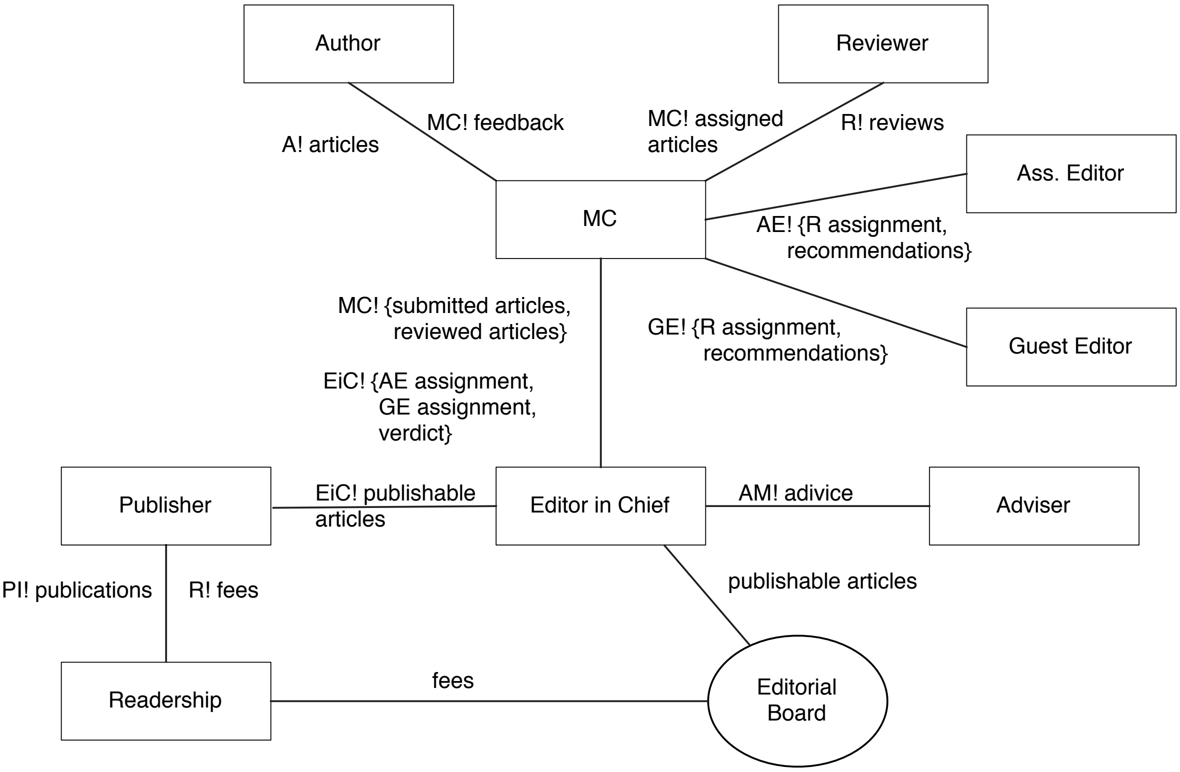

Figure 9. Expert Systems as solution of the Editorial Board problem

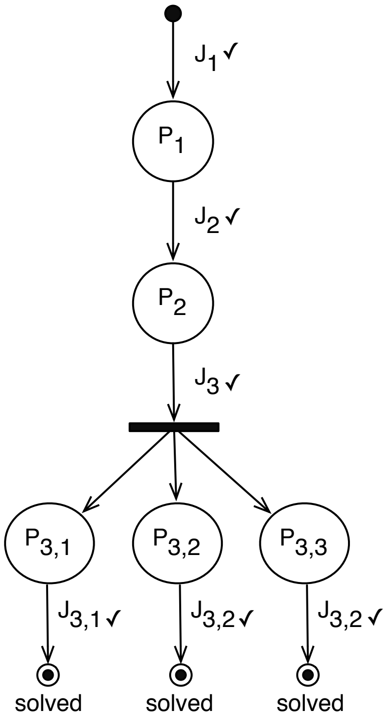

Figure 10. Summary of POE analysis