Building on CHASM: A Study of Using Counts for the Analysis of Static Models of Processes

Keith Thomas Phalp and Martin Shepperd
{kphalp, mshepper}@bournemouth.ac.uk,
Empirical Software Engineering Research Group,
Department of Computing,
Bournemouth University,
Poole, Dorset, BH12 5BB, UK

Abstract

Process modelling is gaining increasing acceptance by software engineers as a useful discipline to facilitate both process understanding and improvement activities. This position paper builds upon previous work reported at the 1997 ICSE workshop on process models and empirical studies of software engineering (Phalp and Counsell 1997). In the previous paper, we argued that simple counts could be used to support analysis of static process models. We also illustrated the idea with a coupling measure for Role Activity Diagrams, a graphical process modelling notation adapted from Petri Nets. At that time only limited empirical work had been carried out, based upon a single industrial study, where we found high levels of coupling in an inefficient process (a more thorough description may be found in (Phalp and Shepperd 1999)). We now summarise a more recent study, which uses a similar analysis of process coupling again based on simple counts. In the study, we compared ten software prototyping processes drawn from eight different organisations. We found that this approach does yield insights into process problems, which could potentially be missed by qualitative analysis alone. This is particularly so when analysing real world processes, which are frequently more complex than their text book counterparts. One notable finding was that despite differences in size and domain, role types across the organisations exhibited similar levels of coupling. Furthermore, where there were deviations in one particular role type, this led the authors to discover a relationship between project size and the coupling levels within that type of role. Given the simplicity of our approach and the complexity of many real world processes we argue that quantitative analysis of process models should be considered as a process analysis technique.

Keywords: process model, Role Activity Diagram, measurement, case study, process improvement

1 Introduction: Why Counts on Process Models

The upstream activities of software development — for example identifying customer requirements — are often considered as the most important, in terms of cost, and yet paradoxically are the least understood aspect of software projects. Business process modelling is one solution that is being increasingly used in conjunction with traditional software development (Phalp 1998), often feeding in to requirements and analysis activities (Yourdon 1994) (Booch, Rumbaugh et al. 1999). In addition, research in Systems Engineering for Business Process Change[1], highlights the importance of modelling business processes in evolving and maintaining the legacy systems that support those processes. However, the major use of business process modelling is to attempt to restructure the business process in order to improve some given aspect, e.g., cost or time. This restructuring may be seen either as separate activity or as a pre-cursor to the development of systems to support the new or improved process. Hence, the analysis of these business models is vital to the improvement of the process, and as a consequence to the development of supporting software systems (PROCESS 1997).

[1] SEBPC is a managed research programme within the Systems Engineering Sub-Programme of the IT&CS Programme of the Engineering and Physical Sciences Research Council (EPSRC). The stated objective of SEBPC is: "to release the full potential of IT as an enabler of business process change, and to overcome the disabling effects which the build-up of legacy systems has on such change", (SEBPC 1998).

Business processes are typically described using static (diagrammatic) models and their analysis is typically qualitative, relying upon the experience of the modeller and the application of guidelines or heuristics (Miers 1994). This paper uses measures (counts) to aid analysis and comparison of these static process descriptions. This idea was proposed at the 1997 ICSE workshop on process models and empirical studies of software engineering (Phalp and Counsell 1997) where the authors showed how counts could be applied to Role Activity Diagrams (RADs), a widely used process modelling notation (Ould 1995). The authors suggested that a simple coupling measure could be used to provide useful guidance to the modeller, but noted the need for further empirical work to investigate the utility of the idea.

The empirical study that will now be described used RADs to describe ten prototyping processes across a number of software developers (Chen 1997). The analysis of these processes was aided by the application of the coupling measure proposed by Phalp and Counsell (Phalp and Counsell 1997). (The background to this study is outlined in Section Three). For further details and an introduction to RADs, refer to the original paper or to a fuller description of the original work (Phalp and Shepperd 1999). The remainder of this position paper briefly recaps the counts utilised, illustrates their application with an example RAD from one of the processes studied and concludes with a short discussion of some outstanding issues and the need for further research.

2 Measuring Coupling in Role Activity Diagrams

In very simplistic terms, a RAD comprises a set of interacting roles (e.g., managing, designing and so forth) (Handy 1976). RADs have behavioural perspective, describing the behaviour of groups or individuals, rather than decomposing the process by function or process (Curtis, Kellner et al. 1992). Consequently, the way a process is partitioned into roles and how these roles communicate with one another is of considerable significance.

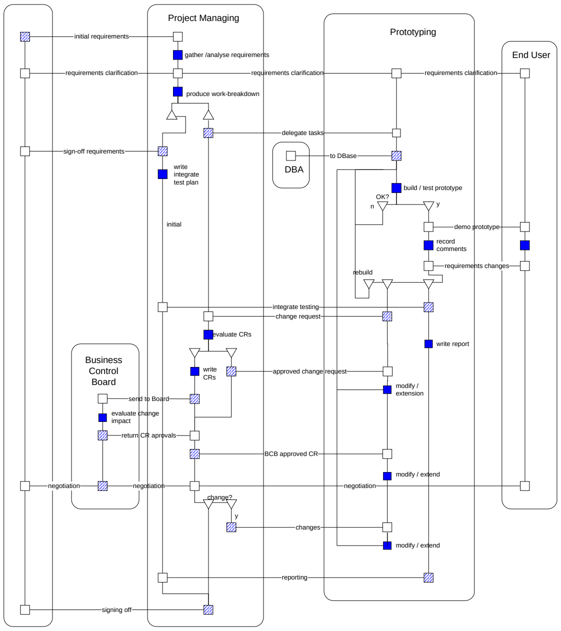

Figure 1: Role Activity Diagram of a Prototyping Process

The coupling measure described is derived from guidelines for process role structuring taken from Ould, who states that as ‘a set, the roles should be loosely coupled, i.e. we should expect few interactions between them’ (Ould 1995). Interactions in RADs may be viewed as shared activities — squares joined by horizontal lines — which act as points of synchronisation moving all roles involved from their current state to their next state (Abeysinghe and Phalp 1997). Phalp and Counsell describe a ratio measure for coupling:

$$CpF_X = (I_X) / (A_X + I_X)$$

where I_X is the count of interactions in role X and A_X is the count of all actions. Note that actions are activities carried out by a role in isolation and are represented as shaded squares. Hence, this ratio has values between zero (only actions and no interactions) and one (only interactions and no actions). A role with neither action nor interaction has no impact on the rest of the process described and is viewed as a separate system with coupling undefined. As an illustration, consider the RAD depicted by Figure 1, and the associated table of values in Table 1.

Roles	Act	Int	CpFx
Business Control Board	1	3	0.75
Project managing	5	13	0.72
Prototyping	6	11	0.65
DBA	0	1	1.00
Customer	0	5	1.00
End user	1	4	0.80

Table 1: Coupling Values for RAD in Figure 1

The benefit of a ratio measure is that it enables comparison between roles of different sizes. It is not, however, our intention to suggest adoption of this single measure, but rather to show the utility of this approach in general. It is likely that other simple counts, such as the number of interactions per role, the size of roles (actions + interactions), would be used along with the coupling measure. Hence, by illustrating what can be gleaned from use of a single simple measure it is hoped that the utility of the approach of using measures in the analysis of process models will be demonstrated.

3 The Empirical Study

The study carried out was part of a larger project to investigate rapid software prototyping processes within a variety of software development organisations (Chen 1997). The aim of the study was to discover commonalties in prototyping processes, which could be used to provide guidelines for the management of prototyping. Ten processes were studied from different domains and with different sizes and characteristics. These ranged in size from 1 to 65 developers and a total number of process participants or actors varying from 4 to 80. Each process was modelled using RADs based on observation, documentation, interviews and workshops. A number of visits were made to each site, to conduct further interviews and to validate and revise the RADs. Although data was obtained from a variety of organisations ranging from airlines to software development within an academic environment, similar roles could be discerned in the processes examined. It is within these roles that general patterns can be found, specifically with respect to coupling.

4 Findings

Table 2 shows the raw data derived from the ten process models and, in addition, information regarding the number of process participants and the size of the development team. Note that for process 6 no data on development team size and participants was available and also that Process 10 did not have a prototyping role comparable to the other teams since it was concerned with prototyping designs for real-time telephone switches.

Process	Roles	Actions	Interact	Prot. role CpF	SysCpF	Participants	DevTeamSize
1	6	13	37	0.65	0.74	8	4
2	4	7	33	0.71	0.83	30	4
3	4	8	22	0.55	0.73	15	1
4	4	12	19	0.5	0.61	5	1
5	4	12	16	0.45	0.57	4	1
6	4	7	24	0.67	0.77		
7	5	7	23	0.56	0.77	15	12
8	7	8	35	0.75	0.81	80	65
9	5	11	29	0.63	0.73	30	30
10	7	8	48			10	5

Table 2: Raw Data from the Process Models

However, rather than examination by process it is more revealing to consider each role type across the various processes. Table 3 shows 3 role types, Customer, Project Managing and Prototyping.

Customer Role				Project Managing Role				Prototyping Role			
Proc	Act	Int	CpFx	Proc	Act	Int	CpFx	Proc	Act	Int	CpFx
1	0	5	1.00	1	5	13	0.72	1	6	11	0.65
				2	2	8	0.80	2	4	10	**0.71**
3	0	7	1.00	3	3	6	0.67	3	5	6	0.55
4	0	5	1.00	4	6	7	**0.54**	4	6	6	**0.50**
5	0	4	1.00					5	6	5	**0.45**
6	0	7	1.00	6	2	7	0.78	6	1	2	0.67
7	0	4	1.00	7	2	6	0.75	7	4	5	0.56
				8	0	3	**1.00**	8	1	3	**0.75**
9	1	5	**0.83**					9	3	5	0.63
10	0	8	1.00	10	2	10	0.83				

Table 3: Raw data by Role Type

An extreme case of similar levels of coupling is that for both customer (shown) and end-user roles all but one role had a coupling factor of one. This contradicts the view that coupling should be minimised. However, one might always expect customers / users to be very highly coupled because from the perspective of the systems engineer the customer is a passive role. Within other types of role, similar levels of coupling could be discerned. That is, coupling levels were consistent with role types being from the same population. Moreover, where roles did not appear to adhere to this pattern, as for project managing and prototyping roles (again shown in Table 3) deviations could be explained by particular circumstances. First, consider the project-managing role. Results show at least two definite outliers; the project managing of processes Four and Eight. In process Eight, (with a high coupling factor) designers undertook a significant amount of managing, and project managers were said to be merely "figureheads", with a limited management role. Hence, this instance of the project-managing role is misleading, and should have been re-classified. In contrast, process Four has a very low value for the coupling factor. However, process Four represents a very small project, where a single developer worked on a project for a customer with whom they had a close working relationship, and hence, communication was minimised.

Finally, consider the prototyping role, which exhibits the largest number of outliers. Figure 2 shows a scatter plot of participant size (Y) against prototyping coupling factor (X); in this case, the stronger of the correlation results.

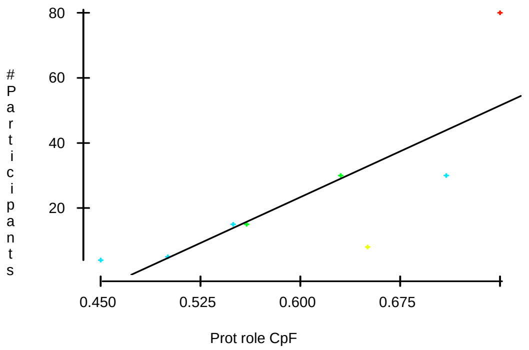

Figure2: Scatter plot of participant size (Y) against prototyping coupling factor (X)

The results show that the prototyping role-coupling factor is correlated using the non-parametric Spearman’s rho (ρ=0.764; significant at the 5% level) with the number of participants and the development team size. This indicates a strong monotonic (possibly non-linear) relationship between the number of process participants and the coupling factor. In other words, larger projects have more mechanisms for communication; suggesting not only that more communication takes place but also that more types of communication are required in order for the project to be managed.

Clearly, this initial study is limited in its coverage; however, it appears that role types exhibit similar coupling levels across organisations, with the size of projects being another factor. Hence, that guidelines could be set to aid identification of outliers, but that these would need to be calibrated both for role type and project size.

5 Conclusions

This paper builds on the idea of applying simple counts to aid analysis of static process models, by applying a coupling measure to Role Activity Diagrams describing ten prototyping processes. The study suggests that the coupling metric may be useful in helping to identify spurious or 'outlier' roles. That is, roles that exhibit particularly high (or low) levels of coupling for their role type within an organisation or site.

However, caution should be exercised in attempting to apply general guidelines for coupling, either across sites or across different role types. For example, the study found a relationship between project size and coupling within prototyping roles. Furthermore, the last thing the authors wish to do is to suggest that the coupling measure described should be adopted as some new process complexity metric. Instead, the usefulness of this simple count, in identifying real world problems, is intended to demonstrate the effectiveness of the general strategy of applying counts to static process models.

The authors recognise the need for further work to assess the usefulness of such metrics in restructuring business processes. However, it is felt that the preliminary work described suggests that there is merit in such further research. Hence, the paper supports the general proposition, that there is merit in applying simple counts to complement traditional forms of business process analysis.

References:

Abeysinghe, G. A. and K. T. Phalp (1997),"Combining Process Modelling Methods." Information and Software Technology **39**(2).

Booch, G., J. Rumbaugh, et al. (1999). The Unified Modeling Language User Guide, Addison-Wesley.

Chen, L. (1997). An Empirical Investigation into Management and Control of Software Prototyping, PhD Thesis. Department of Computing, Bournemouth University.

Curtis, B., M. I. Kellner, et al. (1992). "Process Modelling." Communications of the ACM **35**(9): 75-90.

Handy (1976). On Roles and Interactions. Understanding Organisations, Penguin.

Miers, D. (1994). Use of Tools and Technology within a BPR Initiative. Business Process Re-engineering: myth and reality. C. Coulson-Thomas, Kogan Page.

Ould, M. A. (1995). Business Processes: Modelling and Analysis for Reengineering and Improvement, Wiley.

Phalp, K. T. (1998). "The CAP Framework for Business Process Modelling." Information and Software Technology **40**(13).

Phalp, K. T. and S. J. Counsell (1997). Counts and Heuristics for the Analysis of Static Models. ICSE'97 Workshop on Process Modelling and Empirical Studies of Software Engineering, Boston, 18 May.

Phalp, K. T. and M. Shepperd (1999). Analysing process models quantitatively. EASE'99, Empirical Aspects of Software Engineering, Keele University.

PROCESS (1997). PROCESS: Modelling and Mapping the Business Process. EPSRC project, home page at: http://www.staff.ecs.soton.ac.uk/~ph/process.html.

SEBPC (1998). Systems Engineering for Business Process Change. Homepage at:
http://www.staff.ecs.soton.ac.uk/~ph/sebpc/.

Yourdon, E. (1994). Object-Oriented Systems Design: An Integrated Approach, Prentice Hall.