ELSEVIER

The Journal of Systems and Software 52 (2000) 105–112

The Journal of Systems and Software

www.elsevier.com/locate/jss

Quantitative analysis of static models of processes

Keith Phalp *, Martin Shepperd

Empirical Software Engineering Research Group, Department of Computing, Bournemouth University, Poole House, Talbot Campus, Fern Barrow, Poole, Dorset BH12 5BB, UK

Received 18 November 1999; accepted 18 November 1999

Abstract

The upstream activities of software development projects are often viewed as both the most important, the least understood, and hence the most problematic. This is particularly noticeable in terms of satisfying customer requirements. Business process modelling is one solution that is being increasingly used in conjunction with traditional software development, often feeding in to requirements and analysis activities. In addition, research in Systems Engineering for Business Process Change, [1] highlights the importance of modelling business processes in evolving and maintaining 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. The analysis of these business models is, therefore, vital to the improvement of the process and the development of supporting software systems. Supporting this analysis is the focus of this paper. Business processes are typically described with static (diagrammatic) models. This paper proposes a quantitative approach to aid analysis and comparison of these models. This is illustrated using the process-modelling notation, Role Activity Diagrams (RADs). We studied 10 prototyping processes across a number of organisations and found that roles of the same type exhibited similar levels of coupling across processes. Where roles did not adhere to tentative threshold values, further investigation revealed unusual circumstances or hidden behaviour. Notably, analysis of the prototyping roles (which exhibited the greatest variation in coupling), found that coupling was highly correlated with the size of the development team and the number of participants. This suggests that prototyping in large projects had a different process to that for small projects and required more mechanisms for communication. We conclude that counts (measures) may be useful in the analysis of static process models. © 2000 Elsevier Science Inc. All rights reserved.

Keywords: Process modelling; Role activity diagrams; Measurement; Case study; Process improvement

1. Introduction

Software developers are becoming aware of the need to model the business processes of their clients or customers (Phalp, 1998). This modelling is important because the software being developed should support those business processes, so an important prerequisite is to understand the business needs and context for the proposed system. In addition, the output from business modelling may also be used within the software development process. For example, Yourdon notes how strategic (business) modelling is used as an input to object-oriented analysis (Yourdon, 1994). A further use of business models is within legacy systems. Here the client intends to make changes to a business process supported by an existing system or systems. It is suggested that by understanding the relationship between the business process and the supporting system proposed changes can be more efficiently gauged and managed (PROCESS, 1997). Consequently, a number of researchers have attempted to model both business processes and legacy systems, and construct mappings between them (SEBPC, 1998). This mapping is then used in order to predict how changes to the business process affect the system, and consequently support its evolution.

Software process modelling has been used within software engineering for a number of years, in order to better understand, manage and control the development process (Potts, 1984). The description of customer

* Corresponding author. Tel.: +44-1202-595133; fax: +44-1202-595314.

E-mail addresses: kphalp@bournemouth.ac.uk (K. Phalp), mshepper@bournemouth.ac.uk (M. Shepperd).

[1] Systems Engineering for Business Process Change is a UK research programme funded by the Engineering and Physical Sciences Research Council.

0164-1212/00/$ - see front matter © 2000 Elsevier Science Inc. All rights reserved.
PII: S0164-1212(99)00136-3

processes, however, presents software engineers with a new audience, requiring different approaches and the use of different notations and techniques. For example, if models are to be used in order to describe and validate business needs, then it is important that they be couched in terms that are meaningful to the customer. Thus, it seems sensible to use the kind of models that have been successful within business process re-engineering. (The choice of what kind of model to use is, of course, one that has fuelled a great deal of debate. A discussion of these issues can be found in Phalp (1998).)

Despite the existence of many formal process modelling notations, the majority of the business reengineering community use simple diagrammatic modelling techniques (Miers, 1994). These techniques allow the modeller to discuss and validate process models with both users and process owners, many of whom are not prepared to invest their time in understanding more complex representations. As a result, analysis of processes often consists solely of inspection of diagrams. Typically, this analysis will be guided by the application of heuristics, the experience of the modellers and their knowledge of the particular business domain (Ould, 1995). Analysis can be time consuming and the conclusions are frequently heavily dependent upon the skill of the modeller.

This paper proposes that simple measures of process diagrams can be used to complement and guide expert analysis of process models. We believe that this may be of interest particularly when the processes are complex or expert modellers are unavailable. To illustrate this idea, the paper uses the notation of role activity diagrams (RADs) (Ould, 1995) which is described in the next section. Note that the aim of this paper is to illustrate the utility of a quantitative approach to aid the analysis of static business process models, not to promote RADs, nor the specific measures of RADs suggested. The paper also suggests how various published heuristics for evaluating processes, such as minimising coupling (Ould, 1995), can be supported by associated simple quantitative analysis and outlines the various counts and measures that we utilise. Results from a case study follow. These results suggest that a simple quantitative approach can support the investigation of business processes and is complementary to the usual qualitative means of analysis.

2. An overview of RADs

RADs were originally developed for software process modelling (Ould and Roberts, 1986). The notation reflects the move away from the functional depiction of organisations, to the examination of the behaviour and interactions of individuals or groups (Handy, 1976). RADs have had extensive use and exposure within process modelling and reengineering community. Miers (1994) describes RADs as 'the most powerful method of representing the degrees of freedom, or limits of empowerment offered to workers within the business'.

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

Fig. 1 illustrates a RAD depicting a hypothetical process for a design project, taken from Ould (1995). A role (depicted as a rounded rectangle) groups activities together which might be carried out by a person, group or machine (an actor or an agent). There are three roles in this process model, namely divisional director, project manager and designer.

Actions (indicated by shaded squares) allow a role to move from its current state to a new state. Examples of actions in Fig. 1 include 'prepare a plan' and 'choose a method'. Roles act in parallel, and communicate and synchronise through interactions (shown as unshaded squares joined by a horizontal line). 'Agree TOR for a project' is an example of such an interaction. Interactions are like shared events, in that all roles involved move from their current state to the next state as a result of the interaction. Some authors denote the 'driving' or initiating role of an interaction with a cross-hatched square and this convention is followed within Fig. 1. Hence, a divisional director drives the interaction to agree the terms of reference with a project manager. Vertical state lines joining actions and interactions show the thread of control within a role. A role has constructs to depict concurrent or parallel behaviour, known as part-refinement, shown by a point-up triangle. Choice, known as case-refinement, is shown by a point-down triangle.

Note that roles are like *type*s or *classes* in that they describe a particular kind of behaviour, but are *not* instances of that behaviour. There may be a number of such roles acting in parallel at any given time. For example, in a retail outlet, there might be a number of customer instances and a number of cashier instances. Similarly, a single role may be acted out by a number of different people at different times.

3. Quantitative analysis of RADs

One approach to analysis of process models involves the use of heuristics such as those proposed by Ould (1995) for RADs. In order to facilitate more objective application of these heuristics, various counts have been identified to expedite this form of analysis (Chen, 1997; Phalp, 1998). Consider a familiar concept for software

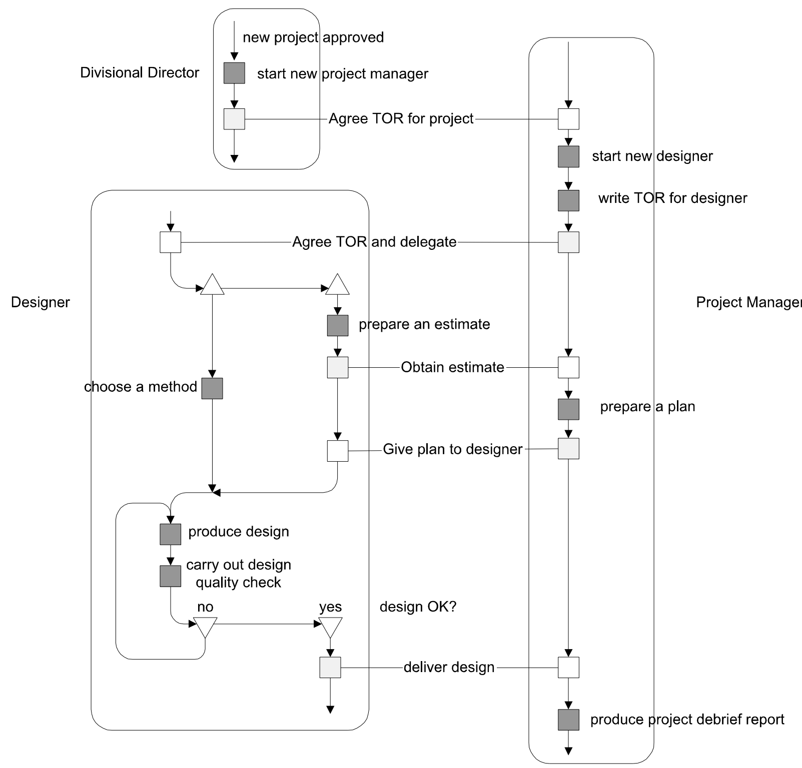

Fig. 1. Example role activity diagram.

engineers: coupling. Ould argues that within business processes – as with software – it is advantageous to minimise coupling. It is thus necessary to understand how coupling is manifested in RADs and to consider whether coupling heuristics are appropriate for business processes. Ould states that

> As a set, the roles should be loosely coupled, i.e., we should expect few interactions between them.

The activity 'carry out design quality check', performed by the designer role in Fig. 1, is internal to that role, and involves no communication with any other role. These internal activities are known as actions within RADs. In contrast, the interaction 'give plan to designer' is a communication between two roles, in the case of the designer example, between the designer and the project manager roles. Counts of these actions and interactions, (of each action or interaction square), form the basis of our proposed RAD coupling measure. An interaction between role X and role Y is, therefore, counted as a separate interaction for each role, i.e., it represents two interactions, since an interaction square is counted in each role. In other words, role X interacts with Y and Y interacts with X, hence two interactions. The 'role-coupling factor' (CpF) of X is calculated by forming the following quotient

$$\mathrm{CpF}_X = (I_X)/(A_X + I_X),$$

where I_X is the count of interactions in role X and A_X is the count of all actions, again, within X. If a role has only actions, that is, it engages in no interactions, the coupling factor will be zero. In practice, this is highly unlikely, since the role would play no part in the remainder of the business process. Similarly, if the role has no actions and only interactions (it is viewed as passive) then the coupling factor is one. This is relatively common as will be seen from the following case study. It is theoretically possible to have a role with neither interaction nor action. However, such a role would have no impact upon the business process. For such a case, the role is viewed as a separate system with the coupling factor undefined.

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.

As an illustration of how coupling factors can be obtained from a RAD, consider the divisional director role in Fig. 1. It has one interaction and one action, hence, the coupling factor is 1/2. The analysis of the remaining roles is summarised in Table 1.

In this example, the coupling factors are similar for each role. One aim in the design of processes would, therefore, typically be to consider the degree of coupling between roles and explore alternatives. Reducing coupling allows roles to become more autonomous and, because they no longer have to synchronise with other roles, gives them the opportunity to complete their tasks more quickly with less opportunity for delay.

Taking these comments about coupling to the extreme implies that the perfect process model contains a single role. However, this role would contain many unrelated tasks and would thus reduce the cohesiveness of that role. Ould observes of cohesion in RADs

> A role should have high cohesion, that is, the activities that form it should be closely related and collectively have a single purpose.

This implies that the role is purposeful and that processes are designed such that a group of tasks is largely self-contained. A role that had many unrelated tasks (low cohesion) would need to communicate with a greater number of roles in order to further the process and would thus often have high coupling. Roles communicate and synchronise only when necessary, however, some separate groupings (roles) are required to maintain cohesiveness. [2] Hence, though one may wish to minimise coupling, some level of coupling (owing to interaction among roles) is unavoidable, and although a significant motivation of this paper was to support the heuristic to reduce role-coupling, in practice this cannot be considered in isolation. Indeed, coupling can always be minimised by the simple expedient of subsuming all activities within a single role, whereas such a process would be highly undesirable. Suffice to say we do not advocate 'magic number' thresholds, nor the optimisation of one heuristic or design feature to the exclusion of all others.

Furthermore, it is unlikely that any specific thresholds for role-coupling would apply equally across the same application domain or even the same organisation. Different types of organisation, process and even role-type would need to be taken into consideration. Indeed, results from the following case study, which used RADs to model business processes and coupling metrics to aid in the analysis of those models, appear to suggest that different role-types typically exhibit quite different levels of coupling.

[2] Given that cohesion is a semantic construct we have not pursued trying to measure it.

Table 1
Quantitative analysis of coupling in an example RAD

Role	A_X	I_X	CpF
Divisional director	1	1	1/2
Projector manager	4	5	5/9
Designer	4	4	4/8

4. Two case studies

In this section we describe two different case studies in which we have endeavoured to apply our ideas of quantitatively analysing process models. The first study is an analysis of the process of gaining new business for a large developer of telecommunications software. This covers the process from the initial enquiry through bids, to customers placing orders. The organisation concerned suggested examination of this process because it was felt to be inefficient and problematic despite representing a key part of the business. The second case study is based on an analysis of ten rapid prototyping processes derived from eight different organisations. Here the aim was primarily to compare and interpret a complex set of process models that appeared to be loosely related.

4.1. Gaining new business

The company is relatively innovative, with a range of products, which are configured to meet the needs of their clients. These clients are typically telecommunications companies. The division of the company where the work was carried out employs over 500 software engineers, and has a traditional top–down management structure.

In order to study the 'official' or theoretical process a number of documents were examined. In addition, information on the actual process was derived from interviews with process actors. These interviews used semi-structured questions and walkthroughs of the RADs produced from the analysis of documents.

To illustrate our ideas of quantitative analysis we describe our analysis of nine sub-processes that relate to the overall process of gaining new business. A fuller discussion may be found in Phalp and Shepperd (1999). To provide some idea of the complexity of the nine process models, combined, these comprise a total of 101 roles, 52 actions and 245 interactions. The high proportion of interactions to actions is striking and confirms the highly coupled nature of the process. The typical process has 12 roles, 4 actions and 26 interactions.

Table 2 shows that the four largest roles have relatively low coupling factors, compared to the remainder. This is because the majority of other roles have zero actions. However, in other respects the four roles are large in that they contain many activities. The business support role is particularly noteworthy in that it has no less than 14 interactions with other roles.

It is therefore not surprising that process actors felt that the overall process was bureaucratic: 'a paper chase'. These actors estimated that over 50% of their time was spent in chasing signatures (of which 'too many were required'). This problem was exacerbated when it was unclear who was the designated signatory in cases where the original signatory was unavailable. Of all the problems, there appeared to be a consensus that chasing information and particularly signatures was the major cost to time and effort. One process actor likened themselves to 'an autograph hunter', and stated that at times it felt as though the process was designed 'so that nothing could get out'.

This view of the process is supported by Table 3. Here we see that in processes such as bid decision the system coupling factor (SysCpF) is close to unity from both the theoretical and proposal perspective, although curiously not from the support perspective. The process contains almost no action type activities.

Rather than allowing individual roles to take responsibility for activities most have shared responsibility and are carried out 'by committee'. This leads to delays, both in scheduling, and in carrying out the tasks. Indeed, such delays associated with interaction were the motivation for Ould's (1995) suggestion that role-coupling should be minimised. However, it was the huge number of single interaction roles that drew the most comment, these often being accounted for by the need for signatures or authorisations. It seems unlikely that all of the signatures needed (over 25) during the process of going from enquiry to submission of proposal were essential. This could be prioritised and rationalised so that some of the signatories are removed, and so the process user can spend a higher proportion of their time in actually composing the bid proposal.

In summary, this case study suggests an overly coupled process, with a very high degree of communication and interaction between roles. Qualitative investigation of the process, by interview and workshops confirmed this view. Our analysis has also identified significant problems of process perception although these have not been discussed in this paper. Consequently, a redesigned process was recommended which allowed roles far more autonomy in the bidding process, and significantly reduced the coupling and cycle time. We also recommended attention to education and communication for and between the process actors.

4.2. Analysis of prototyping processes

Our second case study 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 overall study was to analyse a range of prototyping processes, in order 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 inter-

Table 2
Four largest roles

Perspective	Process	Roles	Act	Int	Act + Int	CpF
Theoretical	Commercial	Bid manager	12	12	24	0.50
Business support	Bid preparation	Business support specialist	8	14	22	0.64
Theoretical	Order processing	Commercial proposal group	9	10	19	0.53
Theoretical	Bid preparation	Proposal specialist	6	12	18	0.67

Table 3
Comparison of role-coupling and size among processes

Perspective	Process	SysCpF	Ints per role	Mean role Sz
Proposal	Bid preparation	0.786	1.83	2.33
Proposal	Bid decision	0.929	2.89	3.11
Support	Bid decision	0.750	1.33	1.78
Support	Bid preparation	0.800	2.86	3.57
Theoretical	Order processing	0.719	3.83	5.33
Theoretical	Commercial	0.720	2.77	3.85
Theoretical	Bid preparation	0.833	2.22	2.67
Theoretical	New business	0.957	1.83	1.92
Theoretical	Bid decision	0.972	2.69	2.77

Table 4
Raw data from the process models

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

views 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.

Table 3 shows the raw data derived from the 10 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.

However, rather than examination by process it is more revealing to consider each role type, across the various processes. Table 4 shows 3 role types, customer, project managing and prototyping. 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, since, one would expect customers/users to be very highly coupled since 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 5) 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 4 and 8. In Process 8 (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 4 has a very low value for the coupling factor. However, Process 4 represents a very small project, where a single developer worked on a project for a customer with whom they had a close working relationship so communication was minimised.

These findings suggest an oversimplification in analysis, making it necessary to consider both the type of process (and organisation) and the role type. For the moment, however, consider an analysis of the prototyping role. The prototyping role exhibits the largest number of outliers. Indeed, the distribution within the prototyping role brings into question the assumption that the sample represents a population of one role type. In other words, more than one kind of behaviour may be hidden within the single role-type description. An examination of the reasons for using prototyping, and the extent of its use within each process, suggests that within the prototyping role there are different sub-processes taking place. These differences may be attributable to the size of the prototyping teams, the mix of abilities in those teams and the control culture in place. Furthermore, coupling in the prototyping role appears to be the main contributor to the system-coupling factor for each process (see Fig. 2).

First, this finding supports the argument that the other role types examined exhibit similar coupling levels across organisations (since much of the variation may be attributed to the prototyping role). Second, this points to the need for further investigation of the prototyping role. Fig. 3 shows a scatter plot of participant size (Y) against prototyping coupling factor (X).

The prototyping role-coupling factor is found to be correlated with the number of participants using the non-parametric Spearman's rho ($\rho = 0.764$; significant at the 5% level). [3] This indicates a strong monotonic 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

[3] The correlation between role-coupling and development team size was also significant at the 5% level.

Table 5
Raw data by role type

Customer role				Project managing role				Prototyping role			
Proc	Act	Int	CpF_X	Proc	Act	Int	CpF_X	Proc	Act	Int	CpF_X
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				

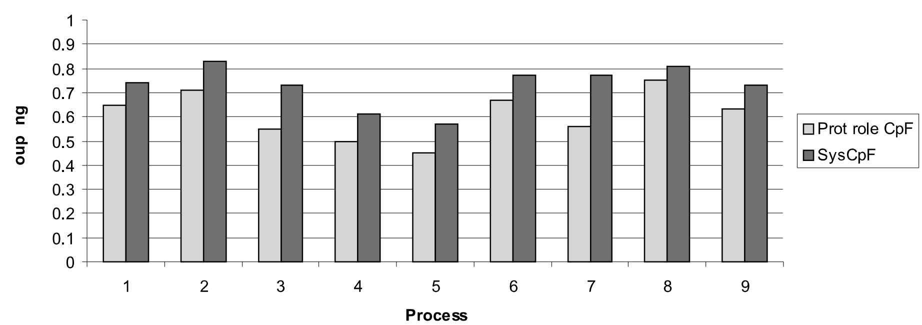

Fig. 2. System coupling and prototyping role-coupling factors.

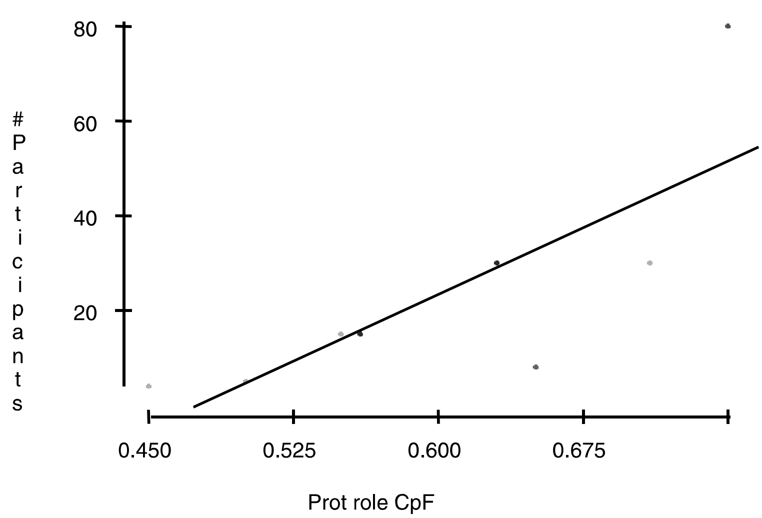

Fig. 3. Scatter plot of participant size (Y) against prototyping coupling factor (X).

more types of communication are required in order for the project to be managed.

This case study finds that the same role types appear to exhibit similar coupling levels across organisations. Outliers in the coupling measures appear to be explained by qualitative evidence gained from process study, and where role types do not appear to adhere to this pattern they conceal different behaviours. For example, variation in the prototyping roles suggested a need for further study concentrating on detailed examination of that role. Study of the prototyping role then revealed a link between the size of development teams (and participants involved) and the extent of coupling in the prototyping role. This suggests that larger processes employ more mechanisms for communication. Hence, prototyping in the large is not the same as prototyping in the small, and different types of behaviour are hidden within the prototyping role.

Clearly, this analysis 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. That guidelines could possibly be set to aid identification of outlier roles, but that these would need to be calibrated both for role type and project size.

5. Conclusions

This analysis has focused upon simple counts of actions and role-coupling. There are, however, other aspects of a RAD that one might wish to explore quantitatively. For example, we have not differentiated between driving interactions and non-driving interactions. This might be useful for identifying roles that tend to be passive and hence more vulnerable to waiting. Other avenues that we do not explore in this paper include part and case refinement, parallel threads and iteration. These are all within-role features that might potentially be analysed.

This paper proposes the idea of applying measures, based on simple counts, to aid the analysis of static process models. The use of such measures allows for the quantification of heuristics to support analysis of business process models. This has been illustrated by describing a coupling measure for RADs. An empirical study of ten prototyping processes was undertaken. The results of the study suggest that the coupling metric may be useful in helping to identify spurious or 'outlier' roles. These are 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 development team size and coupling within prototyping roles. Therefore, although coupling guidelines may be possible, other factors need to be considered.

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. Nor do we argue that quantitative analysis should replace experts.

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

Chen, L., 1997. An empirical investigation into management and control of software prototyping, Ph.D. Thesis, Department of Computing, Bournemouth University.

Curtis, B., Kellner, M.I. et al., 1992. Process modelling. Commun. ACM 35(9), 75–90.

Handy, C., 1976. On Roles and Interactions. Understanding Organisations, Penguin, Harmondsworth.

Miers, D., 1994. Use of tools and technology within a BPR initiative. In: Coulson-Thomas, C. (Ed.), Business Process Re-engineering: Myth and Reality. Kogan Page, London.

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

Ould, M.A., Roberts, C., 1986. Modelling iteration in the software process. In: Proceedings of the Third International Software Process Workshop, Breckenridge, Colorado, USA. IEEE Computer Society Press, Silver Spring, MD.

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

Phalp, K.T., Shepperd, M.J., 1999. Analysing process models quantitatively. In: Proceedings of the Empirical Assessment and Evaluation in Software Engineering, (EASE 99). Keele, UK.

Potts, C. (Ed.), 1984. Proceedings of the First International Software Process Workshop, Egham, Surrey, England. IEEE Computer Society Press, Silver Spring, MD.

PROCESS, 1997. 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: Managed Research Programme of the Engineering and Physical Sciences Research Council (EPSRC). Homepage at: http://www.staff.ecs.soton.ac.uk/~ph/sebpc/, 1998.

Yourdon, E., 1994. Object-Oriented Systems Design: An Integrated Approach. Prentice-Hall, Englewood Cliffs, NJ.